

(11)特許出願公表番号
特表2003-500722
(P2003-500722A)

(43)公表日 平成15年1月7日(2003.1.7)

(51) Int.Cl. ⁷	識別記号	F I	ページ* (参考)	
G 0 6 F 1/00		G 0 6 F 15/00	3 3 0 F	5 B 0 7 6
15/00	3 3 0	9/06	6 6 0 D	5 B 0 8 5

審查請求 未請求 予備審查請求 有 (全 166 頁)

(21) 出願番号	特願2000-620446(P2000-620446)
(86) (22) 出願日	平成12年5月2日(2000.5.2)
(85) 翻訳文提出日	平成13年11月5日(2001.11.5)
(86) 国際出願番号	PCT/US00/11821
(87) 国際公開番号	WO00/072119
(87) 国際公開日	平成12年11月30日(2000.11.30)
(31) 優先権主張番号	09/305, 572
(32) 優先日	平成11年5月5日(1999.5.5)
(33) 優先権主張国	米国(US)

(71)出願人 ラビン・マイケル・オー
RABIN, Michael, O.
アメリカ合衆国, マサチューセッツ州
02138, ケンブリッジ, コンコード アベ
ニュー 243

(71)出願人 シャーシャ・デニス・イー
SHASHA, Dennis, E.
アメリカ合衆国, ニューヨーク州 10012,
ニューヨーク, ブリーカー ストリート
100

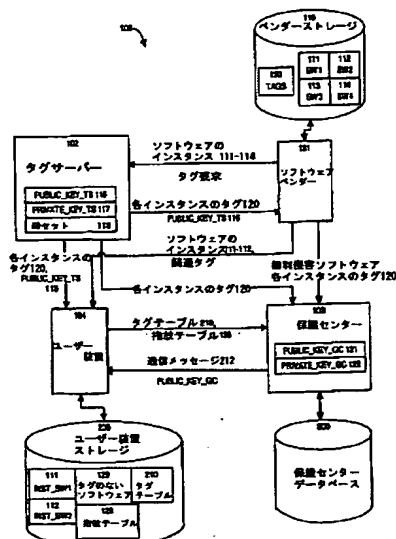
(74)代理人 弁理士 杉本 修司 (外2名)

最終頁に続く

(54) 【発明の名称】 情報保護方法および装置

(57) 【要約】

ソフトウェアのオーナーおよびベンダーが知的所有権を保護し、使用料を請求できるための方法と装置。このシステムはソフトウェアのすべてのインスタンス（事例）ごとに特有のタグを作成する。各ユーザ装置は管理プログラムを実行し、タグを使用して、ソフトウェアオーナーの権利を侵害するソフトウェアのインスタンスが使用されないことを保証する。ソフトウェアのインスタンスをインストールまたは使用する時、管理プログラムは関連タグを確認し、そのタグを格納する。タグ付けされていないソフトウェアをインストールまたは使用すると、管理プログラムはソフトウェアの選択された部分の指紋付けし、その指紋を格納する。ユーザ装置の管理プログラムは定期的に呼び出しするか、または保護センターにより呼び出しされる。保護センターは、現在の呼び出しデータと過去の呼び出し記録とを比較してソフトウェアの無許可使用を検出する。保護センターは管理したソフトウェアのインスタンスの継続使用を有効または無効にして、その呼び出しを完了する。



【特許請求の範囲】

【請求項1】 ソフトウェアの使用を管理するためのシステムであって、ソフトウェアのインスタンスを作成するソフトウェアベンダーと、複数のタグを作成するタグサーバーであって、そのタグはソフトウェアのインスタンスごとに1つ存在し、各タグは、そのタグが関連しているソフトウェアのインスタンスを一意に識別するタグサーバーと、

ソフトウェアのインスタンスを受け取ってインストールし、さらにそのソフトウェアのインスタンスに一意に関連するタグを安全に受け取るユーザー装置であって、ソフトウェアのインスタンスの使用を検出し、ソフトウェアのインスタンスに関連するタグの認証を確認した後、そのソフトウェアのインスタンスの使用を許可する管理プログラムを含むユーザー装置とを備えているシステム。

【請求項2】 請求項1において、ユーザー装置上の前記管理プログラムがタグの認証を確認し、タグテーブル内にそのタグを維持し、そのタグが認証されている場合は前記ソフトウェアのインスタンスを維持し、ソフトウェアに関連するタグが認証されていない場合はソフトウェアのインスタンスを拒絶するシステム。

【請求項3】 請求項2において、前記管理プログラムがタグ内のハッシュ関数値を確認して、そのタグが認証されているかどうか、さらに前記ソフトウェアのインスタンスに適正に関連付けされているかどうかを決定するシステム。

【請求項4】 請求項2において、前記タグがデジタル的に署名され、前記管理プログラムがタグのデジタル署名を確認することによりタグの認証を確認するシステム。

【請求項5】 請求項1において、タグサーバーにより作成された複数のタグのそれぞれが、少なくとも1つのソフトウェア名と、ソフトウェアのインスタンスの固有の番号と、ソフトウェアのインスタンスの各部分のハッシュ関数値を含んでいるシステム。

【請求項6】 請求項5において、前記ソフトウェアのインスタンスの固有の番号が疎セットから選択されているシステム。

【請求項7】 請求項5において、各タグが前記管理プログラムの固有の識

別子をさらに含んでいるシステム。

【請求項8】 請求項7において、前記管理プログラムが、タグ内の管理プログラムの前記固有の識別子がユーザー装置上の管理プログラムの識別子と同一であることを確認するシステム。

【請求項9】 請求項1において、各タグが、そのタグに関連した前記ソフトウェアのインスタンスの各部分で計算された少なくとも1つの指紋を含んでいるシステム。

【請求項10】 請求項9において、前記管理プログラムにより、タグに関連したソフトウェアのインスタンスが、そのソフトウェアのインスタンスに関連したタグ内に含まれる少なくとも1つの指紋に対する同一位置指紋検査を満たすことを確認するシステム。

【請求項11】 請求項10において、前記同一位置指紋検査が、ソフトウェアのインスタンスの使用の前、使用中、および使用後において少なくとも1回前記管理プログラムにより実行されるシステム。

【請求項12】 請求項9において、各タグが、少なくとも1つの指紋を計算できる値を含む少なくとも1つの位置のリストをさらに含み、前記管理プログラムにより、各タグに関連したソフトウェアのインスタンスが、前記少なくとも1つの位置のリスト内で指定された位置におけるソフトウェアに関連する少なくとも1つの指紋に対する同一位置指紋検査を満たすことを確認するシステム。

【請求項13】 請求項1において、前記ソフトウェアのインスタンスによりすべてのデータファイルがアクセスされ、そのアクセスを実行するソフトウェアのインスタンスに関連する情報が前記データファイルに関連する位置に格納されているシステム。

【請求項14】 請求項13において、ソフトウェアのインスタンスに関連する前記情報が、そのソフトウェアのインスタンスに関連するタグであるシステム。

【請求項15】 請求項13において、ソフトウェアのインスタンスに関連する前記情報が、そのソフトウェアのインスタンスにより実行される変更の時刻であるシステム。

【請求項16】 請求項13において、前記アクセスを実行するソフトウェアのインスタンスに関連する前記情報が、前記管理プログラムだけがアクセス可能な安全な位置に書き込まれているシステム。

【請求項17】 請求項16において、前記管理プログラムが、ソフトウェアのインスタンスが1つのデータファイルに関連した位置に格納された関連情報を有するデータファイルにアクセスを試みる時を確認し、さらに前記管理プログラムが、前記格納された関連情報が現在アクセスを試みるソフトウェアのインスタンスに関連する情報であることを確認するシステム。

【請求項18】 請求項16において、前記管理プログラムが偽造できないハッシュ関数を使用して、現在アクセスを試みているデータファイルに関連する位置に格納された関連情報を確認するシステム。

【請求項19】 請求項1において、保護センターが、タグ付けされたソフトウェアデータベースと、確認プログラムとを含み、

前記保護センターが呼び出し手順によって定期的にユーザー装置と通信してユーザー装置からタグを受け取り、前記タグはユーザー装置で使用するタグ付けされたソフトウェアのインスタンスに関連付けされ、前記確認プログラムがタグ付けされたソフトウェアデータベースと対照してユーザー装置から受け取った各タグを検査して、タグが少なくとも1つの使用管理ポリシーに適合していることを保証し、さらに前記確認プログラムがユーザー装置に継続メッセージを返し、その継続メッセージが、ユーザー装置上の各タグに関連したソフトウェアのインスタンスに対してその後の動作を指示するものであり、

ユーザー装置上の前記管理プログラムが前記継続メッセージの認証を確認し、認証した場合は前記継続メッセージ内に指示されたその後の動作を実行するシステム。

【請求項20】 請求項19において、ソフトウェアベンダー、タグサーバーおよび保護センターの少なくとも1つが、ソフトウェアベンダー、タグサーバーおよび保護センターの内の少なくとも1つの他のものと結合されているシステム。

【請求項21】 請求項19において、連続した呼び出し手順間の最大許容

時間間隔が、ユーザー装置内で経過した時間、ユーザー装置が起動された回数、およびユーザー装置の使用程度の組合せの少なくとも1つにより決定されているシステム。

【請求項22】 請求項21において、最後の呼び出し手順後の前記最大許容時間間隔の終了以前に、ユーザー装置が前記保護センターによる呼び出し手順の実行に失敗したときは、そのユーザー装置が一定の期間無効にされるシステム。

【請求項23】 請求項21において、最後の呼び出し手順後の前記最大許容時間間隔の終了以前に、ユーザー装置が前記保護センターによる呼び出し手順の実行に失敗したときは、特定のソフトウェアのインスタンスの使用が一定の期間拒絶されるシステム。

【請求項24】 請求項19において、呼び出しが発生するのが、ソフトウェアのインスタンスがユーザー装置上で最初に使用される時であるシステム。

【請求項25】 請求項19において、呼び出しが発生するのが、前記保護センターからの要求によるものであるシステム。

【請求項26】 請求項19において、前記継続メッセージ内のタグテーブルのハッシュ関数値が、ユーザー装置からの呼び出しメッセージ内で送られたタグテーブル内のハッシュ関数値と同一であることを確認することにより、前記管理プログラムが継続メッセージの認証をテストするシステム。

【請求項27】 請求項26において、前記管理プログラムが、継続メッセージ内のデジタル署名が前記保護センターにより作成されたことを確認することにより、前記継続メッセージの認証がテストされるシステム。

【請求項28】 請求項19において、前記保護センターに対する呼び出しメッセージの後に続く継続メッセージを受信しないユーザー装置は、前の呼び出しメッセージに対するキャンセルコマンドを用いて呼び出しメッセージを再送信するシステム。

【請求項29】 請求項19において、少なくとも1つの使用管理ポリシーが、少なくとも1つのタグに関連付けされた少なくとも1つの個々のソフトウェアのインスタンスに関連付けされているシステム。

【請求項30】 請求項19において、少なくとも1つの使用管理ポリシーが、呼び出し手順中に保護センターが通信する全体のユーザー装置に関連付けされているシステム。

【請求項31】 請求項19において、少なくとも1つの使用管理ポリシーが、呼び出し手順の間に保護センターが通信するユーザー装置の個々のユーザーに関連付けされているシステム。

【請求項32】 請求項19において、少なくとも1つの使用管理ポリシーが、呼び出し手順中に保護センターが通信するユーザー装置の使用履歴に関連付けされているシステム。

【請求項33】 請求項19において、前記保護センターが、各ユーザー装置上の各ソフトウェアのインスタンスの関連する各タグに対する、タグ付けされたソフトウェアデータベースのタグデータ構造体を保持するシステム。

【請求項34】 請求項33において、各タグデータ構造体がソフトウェアのインスタンスのタグと、前記ソフトウェアのインスタンスに関連する使用管理ポリシーと、呼び出し記録に対する参照の収集物とを含むシステム。

【請求項35】 請求項34において、呼び出し記録の前記収集物内の各呼び出し記録が、1つの呼び出し手順に関する情報を表し、また前記呼び出し手順に関連する継続メッセージが、少なくとも1つの呼び出し時刻と、呼び出し手順の間に保護センターに転送されるタグテーブルのヘッダーと、前の呼び出し手順の時刻表示を示す最後の呼び出し時刻と、呼び出し手順の間に保護センターに転送される前記タグテーブルのハッシュ関数値と、ユーザー装置のその後の動作とを含むシステム。

【請求項36】 請求項1において、確認プログラムを含む保護センターをさらに備え、

前記保護センターが呼び出し手順によってユーザー装置と定期的に通信して、そのユーザー装置からユーザー装置管理プログラムに対する固有の識別子を受け取り、前記確認プログラムが前記固有の識別子をテストして、多くても1つの管理プログラムがその識別子を有し、前記確認プログラムがユーザー装置に継続メッセージを返し、ユーザー装置上の各タグに関連するソフトウェアのインスタン

スの使用を試みる際に前記継続メッセージがその後の動作を指示し、

前記ユーザー装置の管理プログラムが継続メッセージの認証を確認し、認証された場合は、その継続メッセージ内の動作を実行するシステム。

【請求項37】 請求項36において、前記管理プログラムが呼び出される最初の時に、まれにしか複製されない番号に基づいて、前記管理プログラム識別子が作成されるシステム。

【請求項38】 請求項37において、前記まれにしか複製されない番号は、前記管理プログラムが装置により最初に呼び出される時に発生する非常に正確なクロック値であるシステム。

【請求項39】 請求項37において、前記まれにしか複製されない事象が、保護センターにより提供される番号であるシステム。

【請求項40】 請求項1において、ユーザー装置上で使用されるタグ付けされていないソフトウェアのインスタンスをさらに備え、

前記管理プログラムが前記タグ付けされていないソフトウェアのインスタンスを検出して、前記タグ付けされていないソフトウェアのインスタンス上の指紋処理を実行し、ユーザー装置上の前記指紋処理の結果である指紋を格納するシステム。

【請求項41】 請求項40において、前記ユーザー装置の管理プログラムが、前記装置上で使用されるタグ付けされていないソフトウェアのインスタンス上の指紋処理をさらに実行し、前記ユーザー装置上の指紋テーブル内の指紋処理から得られる前記指紋を格納するシステム。

【請求項42】 請求項41において、前記管理プログラムが指紋を計算した位置を格納するシステム。

【請求項43】 請求項41において、前記指紋が前記ソフトウェアのインスタンスの内容に基づくシステム。

【請求項44】 請求項41において、前記指紋が前記ソフトウェアのインスタンスの挙動の既知のシーケンスに基づくシステム。

【請求項45】 請求項41において、指紋データ構造体と確認プログラムを含む保護センターをさらに備え、

前記保護センターが呼び出し手順によってユーザー装置と定期的に通信して、前記ユーザー装置上で使用されるソフトウェアのインスタンスに対するユーザー装置からの全指紋を受け取り、前記確認プログラムが前記指紋データ構造体に対してユーザー装置から受け取ったすべての指紋を比較し、ユーザー装置上で使用されるソフトウェアのインスタンスが権利侵害のソフトウェアのインスタンスであるかどうかを決定するシステム。

【請求項46】 請求項45において、前記確認プログラムが、前記保護センターの指紋データ構造体内の指紋とユーザー装置から受け取った指紋との間の一致の指定数より多くの一致を検出した場合は、前記確認プログラムが実行すべき報復措置を指定して、継続メッセージをユーザー装置に返し、前記継続メッセージが前記ユーザー装置上で実行すべき報復措置を指示するシステム。

【請求項47】 請求項46において、前記指紋一致処理が少なくとも1つの一般位置または同一位置指紋一致であるシステム。

【請求項48】 請求項46において、前記指紋一致が反転保護センター指紋テーブルを使用するシステム。

【請求項49】 請求項46において、前記報復措置が、ユーザー装置を指定時間無効にすることを指定するシステム。

【請求項50】 請求項46において、前記報復措置が、保護センターの指紋データ構造体内の指紋に一致した指紋に関連するソフトウェアのインスタンスを指定時間無効にすることを指定するシステム。

【請求項51】 請求項46において、前記報復措置が、ユーザー装置の挙動履歴と、ユーザー装置上の特定ユーザーの挙動履歴と、ユーザー装置上の他のソフトウェアの収集物との組み合わせの少なくとも1つに依存しているシステム。

【請求項52】 請求項45において、前記ソフトウェアベンダーが権利侵害のソフトウェアインスタンスのコピーを保護センターに送信し、前記保護センターが前記権利侵害のソフトウェアインスタンス上の指紋を計算し、さらに保護センター上の指紋データ構造体内に前記指紋を格納するシステム。

【請求項53】 ユーザー装置の読み取り可能媒体上で符号化されたタグテ

ーブルデータ構造体であって、前記タグテーブルデータ構造体が、1つのソフトウェアインスタンスに一意的に関連付けされた少なくとも1つのタグを含み、タグテーブル内の前記タグに関連する少なくとも1つの領域を含み、さらにソフトウェアインスタンスに関連する前記タグに関連する使用状況を表す少なくとも1つの領域を含んでいるタグテーブルデータ構造体。

【請求項54】 請求項53において、前記少なくとも1つの領域が、タグに関連する前記1つのソフトウェアインスタンスに対する使用統計を表しているタグテーブルデータ構造体。

【請求項55】 請求項53において、前記タグテーブルを一意的に識別するタグテーブルヘッダをさらに含んでいるタグテーブルデータ構造体。

【請求項56】 請求項53において、前記タグテーブルヘッダがユーザー装置使用統計に関する情報と継続メッセージを含んでいるタグテーブルデータ構造体。

【請求項57】 それぞれが少なくとも1つの名称とソフトウェア内容を有する、ソフトウェアインスタンスを作成するソフトウェア作成機構と、

各ソフトウェアインスタンスがそのソフトウェアインスタンスに固有のタグだけに関連して使用でき、また前記タグが、そのタグに関連付けされた前記ソフトウェアインスタンスに関連する情報の固有の偽造できない収集物であり、さらに前記ソフトウェアの少なくとも1つの名称と、前記ソフトウェアインスタンスの固有の番号と、前記ソフトウェア内容の各部分のハッシュ関数値を含むソフトウェアインスタンスとを備えているソフトウェアベンダー。

【請求項58】 請求項57において、前記タグが、ソフトウェアインスタンスを使用するユーザー装置に関連する前記管理プログラムの識別子を含んでいるソフトウェアベンダー。

【請求項59】 請求項57において、前記タグが、前記タグが関連付けされているソフトウェアインスタンスの各部分の指紋のリストを含んでいるソフトウェアベンダー。

【請求項60】 請求項57において、ベンダーの権利を侵害するソフトウェアを検出し、その権利侵害のソフトウェアのコピーを保護センターに転送して

使用管理を実行し、ユーザー装置上の権利侵害のソフトウェアインスタンスの使用を検出できるようにする権利侵害ソフトウェア検出機構を備えているソフトウェアベンダー。

【請求項61】 請求項60において、ベンダーの権利を侵害するソフトウェアを検出し、その権利侵害のソフトウェアのコピーを保護センターに転送し、前記保護センターが前記権利侵害のソフトウェアインスタンスに関連するすべてのタグを無効にし、前記権利侵害のソフトウェアインスタンスを使用した保護センターにより検出されたすべてのユーザー装置に対し報復措置を送信する権利侵害ソフトウェア検出機構を備えているソフトウェアベンダー。

【請求項62】 ソフトウェアインスタンスと、一意的に関連付けされたタグと、ソフトウェアインスタンスを使用するための要求とを受け取る入力ポートと、

管理プログラムを実行するプロセッサであって、前記ソフトウェアインスタンス使用するための要求を検出し、ソフトウェアインスタンスに関連するタグの認証を確認した後、ユーザー装置での前記ソフトウェアインスタンスの使用を許可するプロセッサとを備えているユーザー装置。

【請求項63】 請求項62において、前記管理プログラムが、前記タグの認証を確認し、タグテーブル内にそのタグを保持し、そのタグが認証されている場合は前記ソフトウェアインスタンスを維持し、ソフトウェアに関連する前記タグが認証されていない場合は前記ソフトウェアインスタンスを拒絶するユーザー装置。

【請求項64】 請求項63において、前記管理プログラムが、前記ソフトウェアインスタント上のハッシュ関数値を計算し、その計算した値を前記タグ内のハッシュ関数値と比較して、前記タグが認証されているかどうか、さらに前記ソフトウェアインスタンスに適正に関連付けされているかどうかを決定するユーザー装置。

【請求項65】 請求項63において、前記タグがデジタル的に署名され、前記管理プログラムがタグのデジタル署名を確認することによりタグの認証を確認するユーザー装置。

【請求項66】 請求項63において、前記タグテーブルがユーザー装置上のストレージ内に格納されたデータ構造体であり、ソフトウェアインスタンスに一意的に関連する少なくとも1つのタグを含み、さらにタグテーブル内の前記タグに関連する少なくとも1つの領域を含むものであって、前記少なくとも1つの領域が前記タグに関連する前記ソフトウェアインスタンスに対する使用状況を表しているユーザー装置。

【請求項67】 請求項62において、前記管理プログラムが、呼び出しポリシーによる規定にしたがって呼び出し手順を要求することを決定し、さらに前記管理プログラムが、前記呼び出し手順を実行してタグテーブル内に格納されたタグの前記使用状況を更新するユーザー装置。

【請求項68】 請求項62において、前記管理プログラムが、タグ付けされたソフトウェアで使用される各データが正当なソフトウェアインスタンスにより作成されていることを確認するユーザー装置。

【請求項69】 請求項67において、前記呼び出し手順の実行中に、前記管理プログラムがユーザー装置に接続された相互接続機構を介してユーザー装置から前記タグテーブルを安全に送信し、前記ユーザー装置に返送される継続メッセージの受け取りを待機するものであり、前記継続メッセージが前記タグテーブル内の各タグに対し実行すべき動作を指示しているユーザー装置。

【請求項70】 請求項67において、前記呼び出し手順の実行中に、前記管理プログラムが、ユーザー装置に接続された相互接続機構を介してユーザー装置からタグテーブルヘッダを安全に送信し、前記ユーザー装置に返送される、前記タグテーブル内の各タグに対し実行すべき動作を指示している継続メッセージの受け取りを待機するユーザー装置。

【請求項71】 請求項62において、前記ユーザー装置上で使用されるタグ付けされていないソフトウェアインスタンスをさらに備え、

前記管理プログラムが、前記タグ付けされていないソフトウェアインスタンスを検出して、前記タグ付けされていないソフトウェアインスタンス上の指紋処理を実行し、前記ユーザー装置上の指紋テーブル内の前記指紋処理から得た指紋を格納するユーザー装置。

【請求項72】 請求項71において、前記管理プログラムが、呼び出しポリシーによる規定にしたがって呼び出し手順を要求することを決定し、さらに前記管理プログラムが、前記呼び出し手順を実行して、ユーザー装置に格納されたタグ付けされていないソフトウェアインスタンスの使用状況を更新するユーザー装置。

【請求項73】 請求項72において、前記呼び出し手順の実行中に、前記管理プログラムが、ユーザー装置に接続された相互接続機構を介して前記ユーザー装置から前記指紋テーブルの一部を送信し、さらに前記ユーザー装置に返送される、前記ユーザー装置に格納された各タグ付けされていないソフトウェアインスタンスに対し実行すべき動作を指示する継続メッセージの受け取りを待機するユーザー装置。

【請求項74】 タグ付けされたソフトウェアデータベースと、内部のプロセッサで実行する確認プログラムとを備える保護センターであって、

前記保護センターが、呼び出し手順を定期的に行い、相互接続を介して、ソフトウェアインスタンスに対するタグを受け取るものであって、前記確認プログラムが、保護センターに保持されている前記タグ付けされたソフトウェアデータベースに対して受け取ったタグを検査し、前記タグが少なくとも1つの使用管理ポリシーに適合していることを確認し、さらに前記管理プログラムが、相互接続機構を介して、前記呼び出し手順の間に保護センターで受け取られた各タグに関連するソフトウェアインスタンスの使用時にしたがうべき動作を指示している継続メッセージを送信する保護センター。

【請求項75】 請求項74において、少なくとも1つの使用管理ポリシーが、少なくとも1つのタグが関連付けされている各ソフトウェアインスタンスに関連付けされている保護センター。

【請求項76】 請求項74において、少なくとも1つの使用管理ポリシーが、前記保護センターがタグを受け取るために通信するユーザー装置に関連付けされている保護センター。

【請求項77】 請求項74において、少なくとも1つの使用管理ポリシーが、前記保護センターがタグを受け取る目的で通信するユーザー装置の個々のユ

ーザーに関連付けされている保護センター。

【請求項78】 請求項74において、前記タグ付けされたソフトウェアデータベース内に、各ユーザー装置上の各ソフトウェアインスタンスに関連する各タグに対するタグデータ構造体を保持し、またタグサーバからのソフトウェアインスタンスに関連する新しく作成されたタグを受信し、さらにユーザー装置から送信されたタグテーブル内のユーザー装置で使用するソフトウェアインスタンスに関連するタグを受信する保護センター。

【請求項79】 請求項78において、各タグデータ構造体が、ソフトウェアインスタンスの少なくとも1つのタグと、ソフトウェアインスタンスの名称と、前記ソフトウェアインスタンスの固有の番号と、前記ソフトウェアインスタンスのハッシュ関数値と、前記ソフトウェアインスタンスの使用管理ポリシーと、前記ソフトウェアインスタンスに関連するタグに関連付けされた呼び出しレコードへの参照の収集物とを含んでいる保護センター。

【請求項80】 請求項79において、呼び出しレコードの前記収集物内の各呼び出しレコードが1つの呼び出し手順に関する情報を表し、さらに前記各呼び出しレコードが少なくとも1つの呼び出し時刻と、前記呼び出し手順の間に保護センターに転送されるタグテーブルのヘッダーと、前の呼び出し手順の時刻表示を示す最後の呼び出し時刻と、前記呼び出し手順の間に保護センターに転送される前記タグテーブルのハッシュ関数値と、前記呼び出し手順に関連する継続メッセージ内に含まれるユーザー装置のその後の動作とを含む保護センター。

【請求項81】 指紋データ構造体と、確認プログラムを実行するプロセッサとを含む保護センターであって、

前記確認プログラムが、ユーザー装置を用いて呼び出し手順を定期的に実行し、相互接続機構を介して、前記ユーザー装置で使用するソフトウェアインスタンスに対する指紋を受信し、さらに前記確認プログラムが、指紋データ構造体に対して受け取った各指紋を検査し、ユーザー装置で使用するタグ付けされていないソフトウェアインスタンスが権利侵害のソフトウェアインスタンスであるかどうかを決定し、権利侵害している場合は、前記確認プログラムが、前記ユーザー装置で実行される報復措置を準備する保護センター。

【請求項82】 請求項81において、すべてのベンダーソフトウェアが指紋付けされ、別のベンダーソフトウェアに関する1つのベンダーソフトウェアの権利侵害が、少なくとも1つの同一位置または一般位置の指紋検査に基づいて検出される保護センター。

【請求項83】 請求項81において、前記確認プログラムが、指紋データ構造体内の指紋と受け取った指紋内の指紋の間の十分な数の一致を検出する場合は、前記確認プログラムが実行すべき報復措置を指定して、継続メッセージを送信するものであり、前記継続メッセージがそのメッセージの受信機で実行される報復措置を指示している保護センター。

【請求項84】 請求項83において、前記十分な数が1つである保護センター。

【請求項85】 請求項83において、前記十分な数が1よりも大きい保護センター。

【請求項86】 請求項85において、前記十分な数が、各一致の重みが一致する指紋に依存する場合の、一致の重み付けされた総和として計算される保護センター。

【請求項87】 請求項83において、前記指紋照合法が一般位置の指紋検査である保護センター。

【請求項88】 請求項83において、前記報復措置が受信機の実行停止を指定する保護センター。

【請求項89】 請求項83において、前記報復措置が、前記指紋データ構造体内の指紋に一致した指紋に関連したソフトウェアインスタンスを無効にすることを、指定する保護センター。

【請求項90】 請求項81において、前記確認プログラムが、相互接続機構を介して、権利侵害のソフトウェアインスタンスのコピーを受け取り、さらに前記タグ付けされていない権利侵害のソフトウェアインスタンスのコピー上の指紋を計算し、前記指紋を前記指紋データ構造体内に組み込んで格納する保護センター。

【請求項91】 特定ベンダーのソフトウェアのコピーを受け取り、前記ソ

フトウェアインスタンスあたり複数タグを作成するタグサーバにおいて、

各タグがそのタグに関連付けられたソフトウェアインスタンスを一意的に識別し、さらに各タグが前記タグに関連する前記ソフトウェアの少なくとも1つの名称と、前記タグに関連する前記ソフトウェアインスタンスの固有の番号と、前記タグに関連する前記ソフトウェアインスタンスの各部分で計算されたハッシュ関数値とを含んでいるタグサーバ。

【請求項92】 請求項91において、前記タグにデジタル的に署名し、そのタグを、意図する受信機に安全に送信するために使用されるデジタル署名機構をさらに含んでいるタグサーバ。

【請求項93】 ソフトウェアインスタンスを作成するステップと、
前記ソフトウェアインスタンスに一意的に関連付けされたタグを作成するステップと、

前記ソフトウェアインスタンスを配信し、前記タグをユーザー装置に安全に配信し、前記ソフトウェアインスタンスと前記関連するタグを前記ユーザー装置で受け取るステップと、

前記ユーザー装置上での前記ソフトウェアインスタンスの使用を検出するステップと、

使用される前記ソフトウェアインスタンスに関連している前記タグのステータスを判定することにより、前記ソフトウェアインスタンスの使用が可能かどうかを決定するステップとを含んでいるソフトウェアの使用を管理するための方法。

【請求項94】 請求項93において、前記タグを作成するステップが、
前記ソフトウェアインスタンスに固有の番号を割り当てるステップと、
前記ソフトウェアインスタンスの内容の各部の第1ハッシュ関数値を計算するステップと、

前記ソフトウェアインスタンスに対し第2ハッシュ関数値を計算するステップであって、前記第2ハッシュ関数値がソフトウェアの前記名称と、前記ソフトウェアインスタンスに固有の番号と、第1ハッシュ関数値とを組み合わせている、第2ハッシュ関数値を計算するステップと、

前記ソフトウェアインスタンスに一意的に関連付けられるタグを計算するステ

ップであって、前記タグがソフトウェアの前記名称と、前記ソフトウェアインスタンスの固有の番号と、第2ハッシュ関数値とを含む、タグを計算するステップとを含んでいる方法。

【請求項95】 請求項94において、前記タグを計算するステップが、前記第2ハッシュ関数に対してデジタル署名関数を適用し、前記タグ内に前記署名を含むことによりデジタル的に署名されたタグを作成する方法。

【請求項96】 請求項93において、前記タグをユーザーに配信するステップが、前記タグをソフトウェアベンダーに安全に配信するステップを含み、ユーザー装置が公開鍵暗号化方式を使用する方法。

【請求項97】 請求項93において、前記ソフトウェアインスタンスを受け取るステップが、前記ユーザー装置で前記ソフトウェアインスタンスを受け取るステップを含み、

前記ユーザー装置でタグを受け取るステップが、

前記ユーザー装置でソフトウェアインスタンスに関連する前記タグを安全に受け取るステップと、

前記ソフトウェアインスタンスに関連する前記タグが署名されているかどうかを決定し、署名されていれば、前記タグ内のハッシュ関数値の署名を確認して、ユーザー装置に前記ソフトウェアをインストールし、また前記ソフトウェアインスタンスに関連する前記タグが署名されていない場合は、ユーザー装置に前記ソフトウェアインスタンスをインストールするステップとを含んでいる方法。

【請求項98】 請求項93において、前記ユーザー装置での前記ソフトウェアインスタンスの使用を検出するステップが、

前記ユーザー装置上の管理プログラムを呼び出して、前記ソフトウェアインスタンスの使用に対するユーザー要求を遮断するステップを含み、

前記ソフトウェアインスタンスの使用を可能にするかどうかの決定のステップが、

呼び出しポリシーに基づいて呼び出し手順が必要かどうかを決定し、必要なら次の3つのステップ、

1) 呼び出し手順を実行し、認証を確認して、ソフトウェアインスタンスに

関連するタグの前記使用管理ポリシーを決定するステップと、

2) 前記呼び出し手順の結果に基づいて前記ユーザー装置内の情報を更新するステップと、

3) 前記タグに関連するステータス情報を検査して、前記タグに関連するソフトウェアインスタンスの使用が可能かどうかを決定するステップとを実行する方法。

【請求項99】 請求項98において、前記呼び出し手順を実行するステップが、

ユーザー装置から前記ソフトウェアインスタンスに関連する前記タグを格納しているタグテーブルを送信するステップと、

ユーザー装置から返送された、前記タグテーブル内の各タグに対し実行される動作を指示する継続メッセージの受信を待機するステップとを含んでいる方法。

【請求項100】 請求項98において、前記継続メッセージがこの装置方向に向けられ、また事象履歴がこの装置の事象履歴に一致することを確認するステップをさらに含んでいる方法。

【請求項101】 請求項98において、前記呼び出し手順を実行するステップが、

ソフトウェアインスタンスに関連する前記タグを含むタグテーブルを受け取るステップと、

前記タグテーブル内の受け取った各タグをタグ付けされたソフトウェアデータベースに対して検査して、前記タグテーブル内のタグが少なくとも1つの使用管理ポリシーに適合していることを確認するステップと、

各タグに関連する前記ソフトウェアインスタンスの使用を検出した時点で、前記ユーザー装置でその後に実行する動作を指示する継続メッセージを送信するステップとを含む方法。

【請求項102】 請求項101において、前記継続メッセージが、前記継続メッセージを送信する宛先の前記管理プログラムの管理プログラム識別子と、

前記継続メッセージが準備された時刻と、

前記装置からの前記呼び出しに付随しているタグテーブルの符号化とを含んでいる方法。

【請求項103】 ユーザー装置上のタグ付けされていないソフトウェアインスタンスの使用を検出するステップと、

ユーザー装置上の前記タグ付けされていないソフトウェアインスタンスに関連する指紋を作成して格納するステップと、

ユーザー装置上の前記タグ付けされていないソフトウェアインスタンスの使用を検出するステップと、

前記タグ付けされていないソフトウェアインスタンスに関連する指紋と権利侵害の指紋の指紋データ構造体とを比較して、前記ソフトウェアインスタンスの使用が正当であるかどうかを決定し、指紋の一致が発見されると前記タグ付けされていないソフトウェアインスタンスの使用を無効とするステップとを含んでいるソフトウェアの使用を管理するための方法。

【請求項104】 請求項103において、ユーザー装置上のタグ付けされたソフトウェアインスタンスの使用を検出するステップと、

ユーザー装置上の前記タグ付けされたソフトウェアインスタンスに関連する指紋を作成して格納するステップと、

ユーザー装置上の前記タグ付けされたソフトウェアインスタンスの使用を検出するステップと、

前記タグ付けされたソフトウェアインスタンスに関連する指紋と権利侵害の指紋の指紋データ構造体とを比較して、前記ソフトウェアインスタンスの使用が正当であるかどうかを決定し、指紋の一致が発見されると前記タグ付けされたソフトウェアインスタンスの使用を無効とするステップとをさらに含んでいる方法。

【請求項105】 請求項103において、ソフトウェアベンダーにより、権利侵害のソフトウェアインスタンスを検出するステップと、

前記権利侵害のソフトウェアインスタンスのコピーを保護センターに提出するステップと、

前記保護センターにおいて、権利侵害のソフトウェアインスタンス上の指紋を計算し、指紋データ構造体内にその指紋を組み込んで格納するステップとをさら

に含んでいる方法。

【請求項106】 ソフトウェアインスタンスを取得するステップと、
前記ソフトウェアインスタンスに名称を割り当てるステップと、
前記ソフトウェアインスタンスに、同一ソフトウェアの別のインスタンスに割り当てられるすべての固有番号と異なる固有番号を割り当てるステップと、
前記ソフトウェアインスタンスの各部分のハッシュ関数値を計算するステップと、

ソフトウェアインスタンスの前記名称と、前記ソフトウェアインスタンスの番号と、最初に計算されたハッシュ関数値との連続に関する第2ハッシュ関数値を計算して、そのソフトウェアインスタンスに固有の署名のないハッシュ関数値を作成するステップと、

鍵を使用して前記署名のないハッシュ関数値に署名して、前記ソフトウェアインスタンスに対する署名されたハッシュ関数値を作成するステップと、

そのソフトウェアインスタンスを一意的に識別する前記ソフトウェアインスタンスに関連付けされたタグを作成するステップであって、前記タグがソフトウェアインスタンスの前記署名されたハッシュ関数と、ソフトウェアインスタンスの前記名称と、ソフトウェアインスタンスの前記固有番号と、ソフトウェアインスタンスの前記署名のないハッシュ関数とを含む、タグを作成するステップとを含む、ソフトウェアインスタンスを一意的に識別するための方法。

【請求項107】 請求項106において、前記ソフトウェアインスタンスを取得するステップと、そのソフトウェアに名称を割り当てるステップとがソフトウェアベンダーによって実行され、また前記ソフトウェアインスタンスに固有番号を割り当てるステップと、前記第1および第2ハッシュ関数値を計算するステップと、前記第2ハッシュ関数値に署名するステップと、前記タグを作成するステップとがタグサーバーにより実行される方法。

【請求項108】 命令によって符号化されたコンピューター読み取り可能媒体であって、プロセッサで読み取りおよび実行されるときに、ソフトウェアインスタンスを使用する要求を検出するステップを実行し、

そのステップでは、前記ソフトウェアインスタンスに対応するタグが、前記ソ

フトウェアの使用を認める関連ステータスを有するかどうか決定し、定期的に呼び出し手順を実行して前記タグの認証を確認し、前記タグに対応する前記ソフトウェアインスタンスが使用管理ポリシーにしたがって使用されることを確認するコンピュータ読み取り可能媒体。

【請求項109】 通信媒体上を搬送波によって伝送される伝播信号であって、

前記信号が、1つのソフトウェアインスタンスに一意的に関連付けされた少なくとも1つのタグを含み、かつタグテーブル内の前記タグに関連する少なくとも1つの領域を含む符号化されたタグテーブルデータ構造体を有し、前記少なくとも1つの領域が前記タグに関連する1つのソフトウェアインスタンスに対する使用制御ステータスを表している伝播信号。

【請求項110】 媒体上を搬送波により伝送される伝播信号であって、前記信号が符号化された継続メッセージを有し、ある動作に関連するソフトウェアインスタンスの使用要求が受信機で検出されたとき、前記継続メッセージが前記伝播信号の受信機で実行される前記動作の指示を含む伝播信号。

【請求項111】 ソフトウェアプログラムの内容に関し偽造できないハッシュ関数値を計算するステップと、

前記偽造できないハッシュ関数の前記結果を前に保持されたハッシュ関数値と比較してその結果が同一かどうかを決定し、それによりソフトウェアが変更していないかどうかを示すステップとを含んでいるソフトウェアプログラムが変化していないことを確認する方法。

【請求項112】 請求項111において、前記オペレーティングシステムが前記偽造できないハッシュ関数値を計算し、前記ソフトウェアプログラムが前記管理プログラムである方法。

【請求項113】 データの内容に関する偽造できないハッシュ関数値を計算して、その値を前に計算したハッシュ関数値と比較することにより、データが変化していないことを確認する方法。

【請求項114】 請求項113において、前記管理プログラムが、前記偽造できないハッシュ関数値と、前記管理プログラムで使用する前記データとを

計算する方法。

【請求項115】 請求項19において、前記保護センターとユーザー装置間の全メッセージが安全な方法で送信されるシステム。

【請求項116】 請求項115において、前記安全な方法が公開鍵暗号化方式を含むシステム。

【請求項117】 請求項38において、さらに、前記まれにしか複製されない数が少なくとも1つのメモリ位置の値に基づいているシステム。

【請求項118】 請求項80の保護センターにおいて、前記装置からの前記継続メッセージ内に記録された最後の呼び出し時刻が、この装置に対する前記保護センターに記録された最新呼び出しの呼び出し時刻に一致するかどうかを検査する保護センター。

【請求項119】 ソフトウェアの使用を管理するためのシステムであって、

ソフトウェアインスタンスを作成するソフトウェアベンダーと、

ソフトウェアインスタンスを受け取ってそれをインストールするユーザー装置であって、かつ管理プログラムを含むユーザー装置と、

前記ユーザー装置上で使用されるタグ付けされていないソフトウェアインスタンスとを含み、

前記管理プログラムが、前記タグ付けされていないソフトウェアインスタンスの使用を検出し、前記タグ付けされていないソフトウェアインスタンス上の指紋処理を実行し、前記ユーザー装置上の指紋処理から得た指紋を格納するシステム。

【請求項120】 請求項119において、前記ユーザー装置の管理プログラムが、前記装置上でタグ付けされていないソフトウェアインスタンスの指紋処理を実行し、前記ユーザー装置上の指紋テーブル内に前記指紋処理から得た指紋を格納するシステム。

【請求項121】 請求項120において、前記管理プログラムが指紋を計算する位置を格納するシステム。

【請求項122】 請求項120において、前記指紋がソフトウェアインス

タンスの内容に基づいているシステム。

【請求項123】 請求項120において、前記指紋がソフトウェアインスタンスの挙動の既知のシーケンスに基づいているシステム。

【請求項124】 請求項120において、前記保護センターが、指紋データ構造体と確認プログラムとをさらに含み、

前記保護センターが呼び出し手順によってユーザー装置と定期的に通信して、前記ユーザー装置で使用するソフトウェアインスタンスに対するユーザー装置からの全指紋を受け取り、前記管理プログラムが前記ユーザー装置から受け取ったすべての指紋を指紋データ構造体に対して比較して、前記ユーザー装置で使用するソフトウェアインスタンスが権利侵害のソフトウェアインスタンスかどうかを決定するシステム。

【請求項125】 請求項124において、前記管理プログラムが、保護センターの指紋データ構造体内の指紋とユーザー装置から受け取った指紋との一致の特定数より多くの一致がある場合は、前記管理プログラムが実行すべき報復措置を指定し、さらに前記ユーザー装置に継続メッセージを返送するものであり、前記継続メッセージが前記ユーザー装置で実行される前記報復措置を指定するシステム。

【請求項126】 請求項125において、前記指紋一致プロセスが少なくとも1つの一般位置または同一位置の指紋一致であるシステム。

【請求項127】 請求項125において、前記指紋一致が反転保護センター指紋テーブルを使用するシステム。

【請求項128】 請求項125において、前記報復措置が、ユーザー装置を特定時間無効にすることを指定するシステム。

【請求項129】 請求項125において、前記報復措置が、保護センターの指紋データ構造体内の指紋に一致した指紋に関連する前記ソフトウェアインスタンスを特定時間無効にすることを指定するシステム。

【請求項130】 請求項125において、前記報復措置が、ユーザー装置の動作履歴と、前記ユーザー装置上の特定ユーザーの動作履歴と、前記ユーザー装置上の他のソフトウェアの収集物の組合せの少なくとも1つに依存しているシ

ステム。

【請求項131】 請求項124において、前記ソフトウェアベンダーが権利侵害のソフトウェアインスタンスのコピーを保護センターに送信し、前記保護センターが前記権利侵害のソフトウェアインスタンスのコピーの指紋を計算し、さらに保護センター上の指紋データ構造体内に前記指紋を格納するシステム。

【請求項132】 装置識別子を検査内に組み込んだ少なくとも1つのソフトウェアインスタンスを作成するソフトウェア作成機構と、

前記ソフトウェアインスタンスを受け取ってインストールするユーザー装置とを含むソフトウェアベンダーであって、

前記検査が、前記組み込まれた識別子と前記ソフトウェアインスタンスを使用する装置の識別子との比較を含み、

前記組み込まれた識別子が前記装置の識別子に等しい場合は、前記ソフトウェアインスタンスを使用でき、等しくない場合は前記装置上の管理プログラムにより報復措置を実行するソフトウェアベンダー。

【請求項133】 請求項132において、前記ソフトウェアベンダーが前記ソフトウェアインスタンスのハッシュのデジタル署名を送信し、

第2試験が前記デジタル署名が認証できるかどうかを決定し、

第3試験が前記署名された値が前記ソフトウェアインスタンスのハッシュに等しいかどうかを決定するものであって、

前記デジタル署名が認証できないかまたは署名された値が前記ソフトウェアインスタンスと異なる場合は、前記装置の管理プログラムが報復措置を実行するソフトウェアベンダー。

【請求項134】 請求項131において、前記装置識別子が前記ソフトウェアインスタンスの内容の最初の部分または最後の部分に組み込まれているソフトウェアベンダー。

【請求項135】 ソフトウェアの使用を管理するための方法であって、

装置識別子を検査内に組み込んだソフトウェアインスタンスを作成するステップであって、 前記検査が、前記組み込まれた識別子と前記ソフトウェアインスタンスを使用する装置の識別子との比較を含むステップと、

前記ソフトウェアインスタンスをユーザー装置に配布するステップと、

前記検査を実行することにより前記ソフトウェアインスタンスの使用が可能かどうかを決定して、前記組み込まれた識別子が前記装置の識別子に等しい場合は前記ソフトウェアインスタンスを使用できるように使用を許可し、等しくない場合は報復措置を実行するステップとを含んでいる方法。

【請求項136】 請求項135において、前記ソフトウェアインスタンスの前記ハッシュのデジタル署名を送信するステップと、

前記デジタル署名が認証できるかどうかを決定するステップと、

前記署名された値が前記ソフトウェアインスタンスの前記ハッシュに等しいかどうかを決定するステップとを含み、

前記デジタル署名が認証できないかまたは署名された値が前記ソフトウェアインスタンスと異なる場合は、前記装置の管理プログラムが報復措置を実行する方法。

【請求項137】 請求項135において、前記装置識別子が前記ソフトウェアインスタンスの最初の部分または最後の部分に配置されている方法。

【発明の詳細な説明】**【0001】****【発明の背景】**

ソフトウェアもしくは情報の著作権侵害は、当該ソフトウェアもしくは情報の作成者または正当な所有者の許可を得ることなく、ソフトウェアもしくは情報を使用し、またはコピーする行為である。著作権侵害は、コンピュータソフトウェア産業分野において最も横行し、この分野ではソフトウェアアプリケーションの不正コピーが頻繁に行われている。アプリケーションは、私的な使用や、再生産、販売用にコピーされる。他のタイプの著作権侵害行為は、音楽の録音、ドキュメントの読み出し可能なバージョン、または電子ブックのような情報をコピーする行為を含む。著作権の侵害によるビジネスの逸失利益額は、年間に何十億ドルに達すると考えられる。

【0002】

ソフトウェアおよび情報技術産業は、著作権侵害の脅威に対してロック方式の使用により対応してきた。ロック方式は、ソフトウェアロック機構、ライセンス、および特殊なハードウェア装置を有し、これがソフトウェア、情報、またはエレクトロニクス装置全体の不正使用を防止する。これらの方式は、不正使用者がソフトウェアを自由にコピーすることを阻止する。

【0003】

ソフトウェアロック機構には多くのタイプがある。例えばメーカは、ソフトウェアプログラムの部分を鍵（キー）を用いて暗号化する。ソフトウェアを購入するユーザーには、ソフトウェアを解読し実行する鍵が与えられる。このようなソフトウェア保護機構の例は、マイクロソフト ウィンドウズ（登録商標）98のようなソフトウェアの購入時に供給される“真正証明”であり、マイクロソフト コーポレーション（レドモンド，ワシントン州）により製作されている。マイクロソフトおよびウィンドウズ98は、マイクロソフト コーポレーションの商標である。真正証明は、個他の製品番号を示す。ソフトウェアのインストール中、製品番号はソフトウェアアプリケーションにより要求され、ユーザーにより正しく入力される必要がある。入力された製品番号は、ソフトウェアアプリケーシ

ョンにより期待された番号に一致する時にアプリケーションのコピーは正当とされ、インストールされ、実行される。入力された番号が正しくなければソフトウェアは正しくインストールされないことになる。

【0004】

ハードウェア著作権保護方式は、あるハードウェア装置をプロセッサに通信ポートを介して取り付ける。これらのタイプのハードウェア装置は、“ dongle ”と呼ばれる。ハードウェア保護方式の例は、米国特許No. 3, 996, 449に記載され、この特許ではコンピュータの作動中にプログラムまたはプログラムの一部が有効であるか否かを判断するための方法が開示されている。このシステムでは、ハッシュ関数が不正防止ハードウェアチェック装置のプログラムテキストと共にユーザー識別コードまたは鍵に用いられる。チェック装置は、プログラムテキストが正しいか否かを知るためにハッシュ関数から得られた値を確認値と比較する。テキストが正しければプログラムは装置で実行される。

【0005】

他のハードウェア関連の方法は、プログラムを実行する各プロセッサに個々の識別子を割り当てる。これによりソフトウェアプログラムは、そのプログラムが割り当てられ、または実行が許可される指定プロセッサ識別子照合を用いて暗号化される。他のプロセッサの識別子は、ソフトウェアに存在せず、従ってソフトウェアは他のプロセッサで作動することはない。当然このようなシステムは、そのシステムが関連付けられていないプロセッサでソフトウェアを実行しようとする時に使用制限できる。番号割り当て機構は、特定のプロセッサ識別番号を持つ1つのソフトウェアに関連付けられた認可ネットワークの使用を通じて管理される。

【0006】

上記の電子ハードウェア、コンピュータソフトウェアアプリケーションおよびデータ保護機構とは別に、音楽録音装置のようなエレクトロニクス装置によりアクセスされる他のタイプの暗号化情報の著作権侵害を防止するために行われているものは余り知られていない。

【0007】

【発明の概要】**従来技術の特性**

ソフトウェアおよび情報の不正使用を防止するための公知技術は、各種の問題を持っている。真正証明または鍵を用いるシステムは、1つの鍵でプログラムの無制限の使用が可能で、鍵のコピーを防止できない点が問題である。従って、ソフトウェアのコピーの所有者は、彼の鍵または真正証明を他者に渡すことが可能であり、この他者は真正証明または鍵を使用することにより、ソフトウェアをインストールし、作動し、または情報にアクセスすることができる。1つの鍵で1つの使用、または1回の実行のみ行う時には、コピーの問題は解決するが、この時には各使用ごとに他の鍵が必要となる。商業的には、ほとんどのプログラムは多数回の使用を必要とする。

【0008】

ソフトウェアロックは、パーソナルコンピュータで解除することが容易である。パーソナルコンピュータの所有者はロックの解除を試みるのに、無制限の特権と時間を持つからである。ハードウェア保護方式は、ハードウェアデザイナーがハードウェア装置の製作の前に保護すべきソフトウェアの性質を知ることを要求されるから、フレキシビリティを欠く。更に異なったハードウェア保護機構を用いる各種のソフトウェアが作動される時には、別個のハードウェア装置を設ける必要がある。カスタムハードウェア製作にかかるコストおよび消費者がハードウェア保護方式を使用困難と考える事実は、ハードウェア保護機構の広汎な展開を妨げる。

【0009】

ハードウェア保護方式は、ソフトウェアを装置から装置へ移すためのフレキシビリティを制限する。ユーザーは、コンピュータ装置を買う前にソフトウェアを買うことができない。ユーザーは購入の時点で装置の識別子が何であることを知らないからである。ハードウェアメーカーは、多くの装置に同じ装置識別子を与えることによりユーザーをごまかすことができる。最後に熟練したハッカーは、リバースエンジニアリング技法でハードウェア装置の識別子を偽造し、または装置が識別子をチェックしないように、ソフトウェアを変更する。

【0010】

本発明の実施形態の特性

本発明は、上記の問題または他の問題を解決する。本発明は、ソフトウェアの所有者、ベンダーまたはディストリビュータ（以下、ベンダーと呼ぶ）がその知的所有権、またはそのソフトウェアにおける他の権利を保護することを可能にする方法と装置を提供する。ソフトウェアは、コンピュータプログラム、テキスト、データ、データベース、オーディオ、ビデオ、イメージ、その他デジタル的にまたは信号として表される情報のようなものを含む広い意味で定義され、上記ソフトウェアは、コンピュータまたは特殊目的の装置上でユーザーによりアクセスされ、または使用される（以下、この装置をユーザー装置と呼ぶ）。本発明は、ソフトウェアのベンダーに対してソフトウェアの使用ごとに支払う方式にすることも可能にする。

【0011】

特に、本発明は、ユーザー装置上のソフトウェアの使用を管理し、また装置が、ソフトウェアに対する権利の正当なベンダーまたは所有者により許可されない方法で、ソフトウェアのインスタンス、つまりユーザーが使用する個々のソフトウェア、を用いることを防止する管理体制のためのシステム方法または装置を提供する。

【0012】

特定のソフトウェアにおけるベンダーの権利は、下記に限定されない、多数の方法で侵害される。ユーザーは、購入したベンダーのソフトウェアをコピーし、これらを他者に与えることが可能であり、当該他者はソフトウェアの最初のユーザーの購入条件に基づいて許されない時にそのソフトウェアを当該他者の装置にインストールする。ある組織は、ベンダーのソフトウェアを購入し、または賃借する。そして、ソフトウェアの特定数のコピーを行い、使用することが許される場合、その特定数を上回るコピーを行う。著作権侵害ベンダーは、正当なベンダーのソフトウェアの不正コピーをつくり、販売する。著作権侵害ベンダーは、正当なベンダーのソフトウェアを改変し、例えばアプリケーションプログラムを再編集し、または歌の名前を変えるかその他の変更を行い、侵害ソフトウェアのコ

ピーを頒布し販売する。

【0013】

本発明は、ソフトウェアにおける正当なベンダーの権利の保護を達成し、これらの権利侵害をソフトウェアのインスタンス、またはインスタンスの一部を暗号化し、ユーザーにアクセス前に解読することを要求することなく、特殊なハードウェア装置、保護装置（ dongle ）、もしくは特殊なプロセッサを要求することなく、またはメーカに ID 番号をハードウェアに設けることをメーカに要求することなく防止する。従って、これらの方式にかかる短所および弱点は、本発明により回避される。更に本発明の方法と装置は、正当なユーザーが正当なベンダーの特定の体制に基づいて用いるソフトウェアをアクセスすることを妨げるため、非良心的な不正使用者が保護機構の利用を試みる時にも、この試みを阻止する。

【0014】

本発明を用いることによりソフトウェアベンダーは、ユーザーに管理された方法で販売し、リースし、または頒布することを望む特定のアプリケーションプログラム、特定のブック、または歌のような特定のソフトウェアを持つ。ユーザーの装置にインストールし、またはその装置で使うことが試みられるソフトウェアの特定のコピーを、以下、ソフトウェアのインスタンスまたはソフトウェアインスタンスと呼ぶ。一般に、ソフトウェアは、以下、使用またはソフトウェアの使用と呼ぶアクセスモードの各々と共にユーザー装置にインストールされ、アクセスされ、または使用される。このようにアプリケーションプログラムである、例えばソフトウェアのインスタンスの使用は、下記に限定されることはないが、このインスタンスをインストールし、読み、コピーし、または実行することを含む。また、テキストの使用は、下記に限定されることはないが、装置上にテキストをインストールし、またテキストを装置の使用により読み、装置上でまたは装置を用いることによりこのテキストの部分をコピーすることを含む。

【0015】

本発明の具体的な実施形態の要素およびステップ

特に、本発明は、ソフトウェアの使用を管理するためのシステムを提供する。上記のシステムは、ソフトウェアのインスタンスを作るベンダーおよびソフトウ

エアのインスタンスを受け入れるタグサーバーを有する。タグサーバーは、ソフトウェアのインスタンス当り1つのタグで複数のタグを作り、またタグはそれが関連付けられたソフトウェアのインスタンスを個々に特定する。ユーザー装置は、ソフトウェアのインスタンスを受け取り、インストールし、そして、ソフトウェアのインスタンスに個々に関連付けられたタグを受け取る。ユーザー装置は、管理プログラムを含み、これがソフトウェアのインスタンスの使用の試みを検出し、ソフトウェアのインスタンスの使用を許可する前に、ソフトウェアのインスタンスに関連付けられたタグの真正を確認する。ユーザー装置の管理プログラムは、タグの真正を確認し、タグをタグテーブルに保持または格納し、またタグが真正である時には、好ましくは格納装置にソフトウェアのインスタンスを保持し、または格納する。管理プログラムは、ソフトウェアに関連付けられたタグが真正でない時にはソフトウェアのインスタンスを拒絶する。

【0016】

タグは、ソフトウェアのインスタンスにとって固有であることが望ましい。真正証明サーバーにより作られるタグは、ソフトウェアのインスタンスの名称、ソフトウェアのインスタンスの個々の番号および／またはソフトウェアのインスタンスの部分におけるハッシュ関数値の少なくとも1つを有する。好ましくは、ソフトウェアのインスタンスの個々の番号は、番号が疎なセットから選ばれる。他の実施形態においては、各タグは更に管理プログラムの個々の識別子を含む。更に他の実施形態においては、各タグはタグに関連付けられたソフトウェアのインスタンスの部分上で作られた少なくとも1つの指紋（電子透かし）を有する。

【0017】

タグが真正であることを確認し決定するために、管理プログラムはタグにおけるハッシュ関数値を確認し、またはタグのデジタル署名を確認することができる。他の実施形態においては、管理プログラムは、タグにおける管理プログラムの個々の識別子がユーザー装置の管理プログラムの識別子と同じであることを確認する。指紋を用いる実施形態においては、管理プログラムはタグに関連付けられたソフトウェアインスタンスがソフトウェアのインスタンスに関連付けられたタグに含まれる少なくとも1つの指紋に対する同じ位置の指紋チェックを満足する

ことを確認する。同じ位置の指紋チェックは、ソフトウェアのインスタンス前、中および後の少なくとも1回において管理プログラムにより行う。

【0018】

指紋を使用する実施形態において、各タグは、少なくとも1つの指紋が作られて、管理プログラムは、各タグに関連付けられたソフトウェアインスタンスの少なくとも位置の少なくとも1つのリストに定められた位置におけるソフトウェアに関連付けられた少なくとも1つの指紋に対して、同じ位置の指紋チェックを満足することを確認する。または上記に代わり一般位置指紋チェックを用いることができる（同じ位置の指紋において、位置の共通シーケンス上の2シーケンスの指紋は、第1シーケンスから第1指紋が第2シーケンスからの第1指紋に一致し、第1シーケンスからの第2指紋が第2シーケンスからの第2指紋に一致し、以下同様であれば一致する。一般位置指紋方式において、指紋の2シーケンスは、第1シーケンスにおける各指紋が第2シーケンスにおける各指紋に一致し、第2シーケンスにおける各指紋が第1シーケンスにおける各指紋に一致する時に一致する。）。タグは、ソフトウェアのインスタンスから分離しているから、本発明は、ソフトウェアを修正することを必要とせずに、ソフトウェアの保護を提供する。

【0019】

発明の他の構成によれば、いずれかのデータファイルがソフトウェアのインスタンスによりアクセスされる時には、必ずアクセスを実施するソフトウェアのインスタンスに関連付けられた情報がデータファイルに関連付けられた位置に格納されている。ソフトウェアのインスタンスに関連付けられた情報は、ソフトウェアのインスタンスにより実行される改変の時点で、ソフトウェアのインスタンスに関連付けられたタグである。好ましくは、アクセスを実行するソフトウェアのインスタンスに関連付けられた情報は、管理プログラムのみがアクセスできる安全な位置に書き込まれる。本発明のこの構成は、共有されるソフトウェアデータを用いるソフトウェアの著作権侵害を追跡するために用いられる。

【0020】

この場合において、ソフトウェアのインスタンスがデータファイルに関連付け

られた位置で格納される関連付けられた情報を持つデータファイル（即ち、共有されるソフトウェアデータ）をアクセスすることを試みる時には、管理プログラムは格納された関連付けられた情報が現在アクセスを試みるソフトウェアのインスタンスに関連付けられた情報であるか否かをテストする。もしこのような情報である場合には、管理プログラムはこのインスタンスが著作権侵害のコピーであると判断する。これを行うには、管理プログラムはアクセスが現在試みられるデータファイルに関連付けられた位置に格納される関連付けの情報を確認するために偽造不能なハッシュ関数を使用することができる。さらに管理プログラムは、最後の改変の時間を使用できる。本発明は、このデータファイルがこの装置におけるソフトウェアインスタンスのタグを持つソフトウェアインスタンスにより書き込まれたかを監視するものであり、その通りであれば、この装置のソフトウェアインスタンスが事実このデータを最後の改変の時点で書いたか否かを監視する。その通りでなければ、同じタグを持つ少なくとも2つのソフトウェアインスタンスが流通し、著作権の侵害が起きたのである。

【0021】

本発明の他の実施形態は、タグの付いたソフトウェアデータベースおよび確認プログラムを持つ保護センターを有する。保護センターは、定期的にユーザー装置と呼び出し手順により通信することにより、ユーザー装置からタグを受け取る。タグは、ユーザー装置に用いられるタグ付きのソフトウェアのインスタンスに関連付けられる。確認プログラムは、タグが少なくとも1つの使用管理ポリシーが遵守されていることを確かめるためにタグ付きのソフトウェアデータベースに対してユーザー装置から受け取られた各タグを検査する。好ましくは、使用管理ポリシーは、少なくとも1つのタグが関連付けられた少なくとも1つの個別のソフトウェアのインスタンスに関連付けられる。確認プログラムは、継続メッセージをユーザー装置に戻す。継続メッセージは、ユーザー装置における各タグに関連付けられたソフトウェアのインスタンスに対して実行すべき動作（措置）を示す。ユーザー装置における管理プログラムは、真正に対する継続メッセージを確認し、真正であれば継続メッセージに示された続行すべき動作を実行する。この方法で保護センターは、タグ使用状況を管理することにより、ユーザー装置のソ

フトウェアに対するアクセスを決定することができる。

【0022】

好ましくは、保護センターとユーザー装置との間のすべてのメッセージは安全な方法で送られ、この安全な方法は公開鍵暗号方式を含む。

【0023】

本発明の他の構成によれば、ソフトウェアベンダー、タグサーバーおよび保護センターの少なくとも1つは、ソフトウェアベンダー、タグサーバーおよび保護センターの少なくとも他の1つと組み合わせられる。

【0024】

本発明の他の構成において、ユーザー装置の管理プログラムが保護センターと通信する時、この処理を呼び出しと呼ぶ。好ましくは、呼び出し手順の間の最大許容時間間隔は、ユーザー装置において経過する時間、ソフトウェアのインスタンスの使用の数と時間、ユーザー装置が電源オンされる度数およびユーザー装置の使用の測定の組み合わせの少なくとも1つによって決定される。ユーザー装置が最後の呼び出し手順からの最大許容時間間隔の終わりの前に保護センターを用いた呼び出し手順を行うことを怠った時には、ユーザー装置は特定時間にわたり機能を停止され、またはソフトウェアのインスタンスの使用は特定時間帯にわたり否定される。好ましくは、ソフトウェアのインスタンスがユーザー装置において初めて用いられる（即ち、アクセス、インストール、またはその他の形で検出される）時には、呼び出しが行われる。または、呼び出しは保護センターからのリクエストにより生じる。

【0025】

本発明の1つの構成によれば、呼び出し中、管理プログラムは、継続メッセージ中のタグテーブルのハッシュ関数値がユーザー装置からの呼び出しメッセージで送られるタグテーブルのハッシュ関数値に等しいことを確認することにより継続メッセージの真正をテストする。継続メッセージ中のデジタル署名を確認することにも用いられる。

【0026】

呼び出しメッセージに続いて、保護センターへの継続メッセージをユーザー装

置が受け取らない時には、ユーザー装置は前の呼び出しメッセージに対するキャンセル指令を用いた呼び出しメッセージを再度送ることができる。この構成によりユーザー装置は、呼び出しを再び試みることができる。

【0027】

保護センターにおいて、使用管理ポリシーが呼び出し手順中に保護センターと通信する全ユーザー装置に関連付けられるか、使用管理ポリシーが呼び出し手順中に保護センターと通信するユーザー装置の個々のユーザーに関連付けられるか、または使用管理ポリシーが呼び出し手順中に保護センターと通信するユーザー装置の使用管理履歴に関連付けられる。

【0028】

本発明の他の構成によれば、保護センターは各ユーザー装置上のソフトウェアの各インスタンスに関連付けられた各タグに対するタグ付きのソフトウェアデータベースのタグデータ構造体を保持する。各タグデータ構造体は、ソフトウェアのインスタスタグ、ソフトウェアのインスタンスに関連付けられた使用管理ポリシーおよび呼び出し記録についての参照の収集物（集合体）を有する。呼び出し記録の収集物中の各呼び出し記録は、1つの呼び出し手順に関する情報を表わす。呼び出し手順に関連付けられた継続メッセージは、少なくとも1つの呼び出し時間、呼び出し手順中の保護センターに移行するタグテーブルのヘッダ、前の呼び出し手順のタイムスタンプを示す最後の呼び出し時間、呼び出し手順中の保護センターに移されるタグテーブルのハッシュ関数値およびユーザー装置が続行する動作を有する。前の呼び出し記録を保存する理由は、1つの装置のみがタグテーブルの特定のヘッダを持つことを保護センターが確認できるようにすることにある。これが行われなければ、各種の物理的装置が使用管理ポリシーに違反して同じソフトウェアインスタンスを共有することが可能となる。

【0029】

保護センターまたはこれと組み合わされた装置においては、保護センターは確認プログラムを有する。この構成によれば、保護センターはユーザー装置からユーザー装置の管理プログラムのための個々の識別子を受け取るための呼び出し手順を介してユーザー装置と定期的に通信する。確認プログラムは、その識別子を

持つのは多くても1つの管理プログラムであることを確かめるために個々の識別子を調べる。そして、確認プログラムは継続メッセージをユーザー装置に戻す。継続メッセージは、ユーザー装置上の各タグに関連付けられたソフトウェアのインスタンスの使用の試みを続行するための動作を示す。ユーザー装置の管理プログラムは、真正に対する継続メッセージを確認し、真正であれば継続メッセージにおける動作を実行する。

【0030】

保護センターのこの実施形態によれば、管理プログラムの識別子は、管理プログラムが最初に作動される時に、ほとんど重複しない番号に基づいて作られる。好ましくは、上記番号は、管理プログラムが装置上で最初に作動される時に生じる極めて正確なクロック値である。上記に代えて、上記番号は保護センターにより与えられてもよい。上記一方または組み合わせにより、メモリ位置の値によって与えられてもよい。

【0031】

本発明の他のシステムによれば、このシステムは、ユーザー装置に用いられるソフトウェアのタグのないインスタンスをも含む。このシステムにおいて管理プログラムは、タグのないソフトウェアのインスタンスの使用を検出し、タグのないソフトウェアのインスタンスにおいて指紋処理を行い、ユーザー装置の指紋処理からの指紋を格納する。ユーザー装置の管理プログラムは、更に装置に用いられるソフトウェアのタグ付きインスタンスにおいて指紋処理を行い、ユーザー装置の指紋テーブルの指紋処理から生じる指紋を格納する。管理プログラムは、指紋が作られる位置を格納する。指紋は、ソフトウェアのインスタンスの内容に基づいてもよい。また、上記に代えて、指紋はソフトウェアのインスタンスの挙動のシーケンスに基づいてもよい。

【0032】

このシステムにおける保護センターの実施形態によれば、保護センターは、指紋データ構造体および確認プログラムを有する。保護センターは、ユーザー装置で用いられるソフトウェアのインスタンスに対してユーザー装置からすべての指紋を受け取るように、呼び出し手順を介してユーザー装置と定期的に通信する。

確認プログラムは、ユーザー装置から受け取ったすべての指紋を指紋データ構造体と比較することにより、ユーザー装置に用いられるソフトウェアのインスタンスがソフトウェアの侵害インスタンスであるか否かを判断する。確認プログラムが保護センターの指紋データ構造体とユーザー装置から受け取られた指紋との間で指定数以上の一致を検出した時には、確認プログラムは実行すべき報復措置を特定し、確認プログラムは継続メッセージをユーザー装置に戻す。継続メッセージは、ユーザー装置で実行される報復措置を示す。

【0033】

ソフトウェアベンダーは、ソフトウェアの侵害インスタンスのコピーを保護センターに送り、保護センターは、ソフトウェアの侵害インスタンスのコピー上の指紋を算定し、保護センターの指紋データ構造体の中に指紋を格納する。

【0034】

このシステムの1つの構成によれば、指紋一致処理は、一般的な位置指紋チェックである。処理を速めるために指紋一致は、反転保護センター指紋テーブルを使用する。

【0035】

報復措置は、ユーザー装置が指定時間にわたり機能を停止されるか、または保護センターの指紋データ構造体の指紋に一致した指紋に関連付けられたソフトウェアのインスタンスが指定時間にわたり機能停止されるかを指定できる。報復措置は、ユーザー装置の挙動履歴の組み合わせ、ユーザー装置における特定ユーザーの挙動履歴およびユーザー装置におけるソフトウェア供給の集合の少なくとも1つによって特定される。

【0036】

本発明の他の実施形態は、ユーザー装置の読み取り可能媒体、例えば装置読み取り可能媒体で符号化されたタグテーブルデータ構造体を提供する。タグテーブルデータ構造体は、ソフトウェアの1つのインスタンスに個々に関連付けられた少なくとも1つのタグを有し、タグテーブルのタグに関連付けられた少なくとも1つの領域およびソフトウェアのインスタンスに関連付けられたタグに関連付けられた使用状況を示す少なくとも1つの領域を有する。少なくとも1つの領域は

、タグに関連付けられたソフトウェアの1つのインスタンスに対する使用統計を示す。タグテーブルは、タグテーブルを個々に識別するタグテーブルヘッダを有する。タグテーブルヘッダは、ユーザー装置の使用統計に関する情報を有し、同様に継続メッセージをも有することができる。タグテーブルは、ユーザー装置で用いられるべきソフトウェアのインスタンスの能力に関する情報を格納するために用いられる。

【0037】

本発明の装置と方法は、それぞれ少なくとも1つの名称とソフトウェアの内容を持つソフトウェアのインスタンスを形成するソフトウェア形成機構からなるソフトウェアベンダーを有する。ソフトウェアの各インスタンスは、ソフトウェアのインスタンスごとのタグに関連してのみ使用できる。好ましくは、タグはタグが関連付けられたソフトウェアのインスタンスに関する個々に偽造不能な情報の収集物であり、ソフトウェアの1つの名称、ソフトウェアのインスタンスの番号、ソフトウェアの内容の一部におけるハッシュ関数値、ソフトウェアのインスタンスが使用されるべきユーザー装置に関連付けられた管理プログラムの識別子、またはタグが関連付けられたソフトウェアのインスタンスの部分の指紋リストの少なくとも1つを有する。

【0038】

本発明の他の実施形態によれば、ソフトウェアベンダーは、ベンダーの権利を侵害するソフトウェアを検出し、侵害ソフトウェアのコピーを保護センターに送ることにより、ユーザー装置で侵害ソフトウェアのインスタンスの試みられた使用を検出するように使用管理を実施できる侵害ソフトウェア検出機構を有する。

【0039】

この実施形態の他の構成によれば保護センターは、侵害ソフトウェアのインスタンスに関連付けられたタグを無効化し、保護センターにより侵害ソフトウェアのインスタンスを使用したことを検出されたユーザー装置に報復措置を送ることができる。

【0040】

本発明の他の実施形態は、ソフトウェアのインスタンスを受け入れ、ソフトウ

エアインスタンスに個々に関連付けられたタグを受け入れ、そしてソフトウェアのインスタンスを使用することのリクエストを受ける入力ボードを含むユーザー装置である。ユーザー装置に含まれるプロセッサは、管理プログラムを実行する。管理プログラムは、ソフトウェアのインスタンスを使用するためのリクエストを検出し、ユーザー装置によるソフトウェアのインスタンスの使用許可の前に、ソフトウェアのインスタンスに関連付けられたタグが真正であることを確認する。管理プログラムは、タグが真正であることを確認し、タグテーブルにおけるタグを格納し、タグが真正であればソフトウェアのインスタンスを保持し、ソフトウェアに関連付けられたタグが真正でなければソフトウェアのインスタンスを拒絶する。

【0041】

ユーザー装置の1つの構成によれば、管理プログラムは、ソフトウェアのインスタンスにおけるハッシュ関数値を算定し、算定値をタグのハッシュ関数値と比較することによりタグが真正であるか否か、またソフトウェアのインスタンスに正しく関連付けられているか否かを判断する。好ましくは、タグはデジタル的にサインされ、管理プログラムはタグのデジタル署名を確認することによりタグが真正であることを確認する。

【0042】

ユーザー装置の中で、タグテーブルはユーザー装置に格納されるデータ構造体であり、ソフトウェアのインスタンスに個々に関連付けられた少なくとも1つのタグを有し、タグテーブルの中のタグに関連付けられた少なくとも1つの領域を有し、この少なくとも1つの領域はタグに関連付けられたソフトウェアのインスタンスに対する使用状況を示す。管理プログラムは、呼び出し手順が呼び出し方式により定められたようにリクエストされることを定期的に決定し、そしてタグテーブルに格納されているタグの使用状況を更改するための呼び出し手順を実行する。

【0043】

管理プログラムは、タグ付きのソフトウェアにより用いられる各データファイルが正当なソフトウェアのインスタンスにより形成されていることを確認する。

【0044】

呼び出し手順の実行中、管理プログラムは、ユーザー装置に結合された相互接続機構を介してユーザー装置からのタグテーブルを安全に送り、ユーザー装置に戻される継続メッセージの受け入れを待つ。上記継続メッセージはタグテーブルの各タグに対して実行すべき動作を示す。また、呼び出し手順の実行中にも、管理プログラムは、ユーザー装置に結合された相互接続機構を介してユーザー装置からのタグテーブルヘッダを安全に送り、タグテーブルの各タグに対して実行されるべき動作を示すユーザー装置に戻される継続メッセージの受け入れを待つ。

【0045】

本発明の他の実施形態は、タグなしのソフトウェアの使用を管理することを可能にする。この実施形態によるユーザー装置は、ユーザー装置に用いられるタグのないソフトウェアのインスタンスを有する。管理プログラムは、ソフトウェアのタグなしのインスタンスを検出し、指紋処理を実施し、ユーザー装置の指紋テーブルにおける指紋処理から得られた指紋を格納する。管理プログラムは、呼び出し手順が呼び出し方式によって定められるようにリクエストされることを決定し、管理プログラムは呼び出し手順を実行することによりユーザー装置に格納されるタグのないソフトウェアのインスタンスの使用状況を更新する。このようにタグのないソフトウェアの管理は、タグ付きのソフトウェアの存在、または管理とは無関係に行うことができる。

【0046】

呼び出し手順を行う時に、管理プログラムは、ユーザー装置に結合された相互接続機構を介してユーザー装置から指紋テーブルの部分を送り、ユーザー装置に格納された各タグのないソフトウェアのインスタンスに対して実行すべき動作を示すユーザー装置に戻された継続メッセージの受け取りを待つ。

【0047】

本発明の他の実施形態によれば、保護センターは、タグ付きのソフトウェアデータベースおよび保護センターのプロセッサで実行する確認プログラムを含むものが設けられている。保護センターは、相互接続機構を介してソフトウェアのインスタンスに対するタグを受け取るための呼び出し手順を定期的に実行する。確

認プログラムは、タグが少なくとも1つの使用管理ポリシーに従うものであることを確かめるために保護センターに保持されるタグ付きソフトウェアデータベースを用いて受け取られた各タグを検査する。確認プログラムは、相互接続機構を介して呼び出し手順中に保護センターにより受け入れられる各タグに関連付けられたソフトウェアのインスタンスの試みられた使用を続行するように動作を示す継続メッセージを送る。

【0048】

この実施形態のある構成によれば、使用管理ポリシーは、少なくとも1つのタグに関連付けられたソフトウェアの各インスタンスに関連付けられる。また、使用管理ポリシーは、保護センターがタグを受け取るために通信するユーザー装置に関連付けられる。使用管理ポリシーは、保護センターがタグを受け取るために通信するユーザー装置の個々のユーザーに関連付けられる。

【0049】

保護センターは、各ユーザー装置のソフトウェアのインスタンスに関連付けられた各タグに対するタグ付きソフトウェアデータベースのタグデータ構造体を保持し、タグサーバーからソフトウェアのインスタンスに関連付けられた新たに作られたタグを受け取り、更にユーザー装置から送られたタグテーブルのユーザー装置で用いられるソフトウェアのインスタンスに関連付けられたタグを受け取る。各タグデータ構造体は、ソフトウェアのインスタンスのタグ、ソフトウェアのインスタンスの名称、ソフトウェアのインスタンスの番号、ソフトウェアのインスタンスのハッシュ関数値、ソフトウェアのインスタンスに関連付けられた使用管理ポリシーおよび上記ソフトウェアのインスタンスに関連付けられたタグに関連付けられた呼び出し記録についての参照の収集物の少なくとも1つを有する。

【0050】

呼び出し記録の収集物における各呼び出し記録は、1つの呼び出し手順に関する情報を表し、呼び出し時間、呼び出し手順中に保護センターに送られたタグテーブルのヘッダ、前の呼び出し手順のタイムスタンプを示す最後の呼び出し時間、呼び出し手順中の保護センターに送られたタグテーブルのハッシュ関数値および呼び出し手順に関連付けられた継続メッセージに含まれるユーザー装置を続行

するための動作の少なくとも1つを有する。

【0051】

この発明による保護センターの他のタイプは、指紋データ構造体および確認プログラムを実行するプロセッサを有する。確認プログラムは、ユーザー装置で用いられるソフトウェアのインスタンスに対する指紋を相互接続機構を介して受け取るようにユーザー装置に対して呼び出し手順を定期的に実行する。確認プログラムは、指紋データ構造体に対して受け取られた各指紋を検査することにより、ユーザー装置に用いられたタグのないソフトウェアのインスタンスがソフトウェアの侵害インスタンスであるか否かを調べ、侵害インスタンスであれば、確認プログラムはユーザー装置に対して実行されるべき報復措置を生成する。

【0052】

ある実施形態においては、すべてのベンダーソフトウェアは指紋が作られ、1つのベンダーソフトウェアによる他のベンダーソフトウェアに対する侵害が一般位置指紋チェックに基づいて検出される。確認プログラムが指紋データ構造体と受け取った指紋の中における指紋との間で十分な数の一致を検出する時には、確認プログラムは実行されるべき報復措置を特定し、確認プログラムは継続メッセージを送る。上記継続メッセージは継続メッセージの受信機に対して実行される報復措置を示す。十分な数の一致は1つ以上であるか、または各一致の重みが一致する指紋によって変化する時には、一致の加重合計値として算定される。

【0053】

この実施形態の他の構成によれば、報復措置は受信機の機能停止、または指紋データ構造体の中の指紋に一致した指紋に関連付けられたソフトウェアのインスタンスが機能停止されるべきことが特定される。

【0054】

他の方法では、保護センターで確認プログラムが相互接続機構を介してソフトウェアの侵害インスタンスのコピーを受け取り、タグのないソフトウェアインスタンスのコピーを受け取り、タグのない侵害ソフトウェアインスタンスのコピー上の指紋を算定し、指紋を指紋データ構造体に入れて格納する。

【0055】

本発明の実施形態は、特定のベンダーソフトウェアのコピーを受け入れ、複数のタグをソフトウェアインスタンス当たり1つの割合で作り、上記各タグが関連付けられたソフトウェアのインスタンスを個々に識別するものであるタグサーバーを有する。好ましくは、各タグは、タグに関連付けられたソフトウェアの名称、タグに関連付けられたソフトウェアのインスタンスの番号、およびタグに関連付けられたソフトウェアのインスタンスの部分において算定されたハッシュ関数値の少なくとも1つを有する。タグをデジタル的にサインし、タグを例えばユーザー装置、保護センター、またはソフトウェアベンダーのような受信機に安全に送るためにデジタル署名機構を用いることができる。

【0056】

本発明にかかる方法は、ソフトウェアの使用を管理するための方法を有する。この方法は、ソフトウェアのインスタンスを作り、ソフトウェアのインスタンスに個々に関連付けられるタグを作るステップを有する。次にソフトウェアのインスタンスを配布し、タグをユーザー装置に安全に配布し、そして、ユーザー装置のタグおよび関連付けられたタグを受け取る。次にユーザー装置のソフトウェアのインスタンスを使用するための試みを検出し、そのソフトウェアのインスタンスを使用するための試みが、使用されるべきソフトウェアのインスタンスに関連付けられたタグのステータスを判断することによって許容されるか否かを判断する。

【0057】

上記の方法において、タグ製作は、ソフトウェアのインスタンスに対する番号を割り当て、ソフトウェアのインスタンスの内容の部分での第1ハッシュ関数値を算定するステップを有する。次に、ソフトウェアのインスタンスに対する第2ハッシュ関数値を算定することを有し、上記第2ハッシュ関数値はソフトウェアの名称、ソフトウェアのインスタンスの番号および第1ハッシュ関数値を組み合わせる。次に、ソフトウェアのインスタンスに個々に関連付けられたタグを算定するステップを有し、上記タグはソフトウェアの名称、ソフトウェアのインスタンスの番号および第2ハッシュ関数値を有する。

【0058】

タグを作るステップは、署名を作るための第2のハッシュ関数値にデジタル署名機能を適用し、署名をタグに含めることによりデジタル的にサインされたタグを作る。

【0059】

タグをユーザー装置に配布するステップは、公開鍵暗号技術を用いてソフトウェアベンダーおよびユーザー装置に安全にタグを配布するステップを有する。

【0060】

ソフトウェアのインスタンスを受け取るステップは、ユーザー装置においてソフトウェアのインスタンスを入手するステップを有する。ユーザー装置のタグを受け取るステップは、ユーザー装置におけるソフトウェアのインスタンスに関連付けられたタグを安全に受け取り、ソフトウェアのインスタンスに関連付けられたタグがサインされたか否かを判断し、サインされている時には、タグのハッシュ関数値の署名を確認し、ハッシュ関数値上の署名が確認されるとユーザー装置上にソフトウェアをインストールし、ソフトウェアインスタンスに関連付けられたタグがサインされていない時には、ユーザー装置上にソフトウェアのインスタンスをインストールするステップを有する。ユーザー装置でソフトウェアインスタンスを使用する試みを検出するステップは、ソフトウェアのインスタンスの使用に対するユーザーのリクエストを阻止するためにユーザー装置に管理プログラムを作動させるステップを有する。ソフトウェアインスタンスを用いる試みが許されるか否かを判断するステップは、呼び出し手順が呼び出し方式に基づいて必要とされるか否かが判断され、必要とされる場合には、呼び出し手順を行うことにより真正であることを確認し、ソフトウェアインスタンスに関連付けられたタグの使用管理ポリシーを決定するステップを有する。また、呼び出し手順の結果に基づいてユーザー装置におけるタグ情報を更新するステップを有し、タグに関連付けられたソフトウェアのインスタンスの使用が許されるか否かを判断するためにタグに関連付けられたステータス情報を検査するステップを有する。

【0061】

呼び出し手順を実行するステップは、ユーザー装置からのソフトウェアのインスタンスに関連付けられたタグを格納するタグテーブルを送るステップと、タグ

テーブルにおける各タグに対して実行されるべき動作を示すユーザー装置に戻される継続メッセージの受け取りを待つステップとを有する。ユーザー装置は、継続メッセージを待つ間、実行のための端末リクエストの処理を継続する。

【0062】

方法の実施形態は、継続メッセージが特定の装置に向かって送られるステップ、およびイベント履歴がこの装置のイベント履歴に対応することを確認するステップを有する。

【0063】

方法の実施形態において、呼び出し手順を実行するステップは、ソフトウェアのインスタンスに関連付けられたタグを含むタグテーブルを受け取るステップと、タグテーブルのタグが少なくとも1つの使用管理ポリシーに従っていることを確かめるためのタグ付きのソフトウェアデータベースに対してタグテーブルに受け取られた各タグを検査するステップを有する。また、各タグに関連付けられたソフトウェアのインスタンスの使用の試みを検出した時にユーザー装置で続行するための動作を示す継続メッセージを送るステップも含まれる。

【0064】

方法の実施形態において、継続メッセージには、継続メッセージが送られる相手の管理プログラムの管理プログラム識別子、継続メッセージが準備された時点および装置からの呼び出しに伴うタグテーブルヘッダの暗号化を含むことができる。

【0065】

ソフトウェアの使用を管理するための方法は、本発明の一部として提供され、ユーザー装置上のタグのないソフトウェアのインスタンスの使用を検出し、ユーザー装置上のタグなしのソフトウェアインスタンスに関連付けられた指紋を作り、格納するステップを有する。続いて、ユーザー装置におけるタグのないソフトウェアのインスタンスを使用することの試みを検出し、ソフトウェアのインスタンスを使用することの試みがタグのないソフトウェアのインスタンスに関連付けられた指紋を侵害指紋の指紋データ構造体と比較することにより有効であるか否かを判断し、指紋が一致する時には、タグのないソフトウェアのインスタンスの

使用を機能停止する。

【0066】

上記の方法は、ユーザー装置上のタグ付きのソフトウェアのインスタンスの使用を検出するステップ、およびユーザー装置のタグ付きのソフトウェアのインスタンスに関連付けられた指紋を作り、格納するステップを有する。ユーザー装置のタグ付きのソフトウェアのインスタンスを使用する試みを検出するステップは、ソフトウェアのインスタンスを使用することの試みが、タグ付きのソフトウェアのインスタンスに関連付けられた指紋を侵害指紋の指紋データ構造体と比較することにより有効であるか否かを判断し、指紋が一致する時には、タグ付きのソフトウェアのインスタンスの使用の機能を停止するステップである。

【0067】

上記の方法は、ソフトウェアベンダーにより侵害ソフトウェアのインスタンスを検出し、侵害ソフトウェアのインスタンスのコピーを保護センターに提出するステップにより補足される。保護センターでソフトウェアの侵害インスタンス上の指紋を作り、指紋を指紋データ構造体に格納するステップも含まれる。この補足的な方法は、タグ付きのソフトウェアの存在に関係がなく、他の実施形態でもあり得る。

【0068】

本発明の他の実施形態は、ソフトウェアのインスタンスを入手し、ソフトウェアのインスタンスに名称を割り当て、そしてソフトウェアのインスタンスに番号を割り当てるステップを持つソフトウェアのインスタンスを個々に識別するための方法を有する。上記番号は、同じソフトウェアの他のインスタンスに割り当てられた番号とは異なることがある。この方法は、ソフトウェアのインスタンスの部分におけるハッシュ関数値を算定し、インスタンスソフトウェアの名称のシーケンスにおける第2ハッシュ関数値、インスタンスソフトウェアの番号、およびソフトウェアインスタンスごとの署名のないハッシュ関数値を作るための第1ハッシュ関数値を算定するステップをも有する。続いて、ソフトウェアのインスタンスに対する署名されたハッシュ関数値を作るための鍵を用いて署名されないハッシュ関数値を署名し、ソフトウェアのインスタンスを個々に識別するソフトウ

エアのインスタンスに関連付けられたタグを作る。上記タグはソフトウェアのインスタンスの署名されたハッシュ関数値、ソフトウェアのインスタンスの名称、ソフトウェアのインスタンスの番号およびインスタンスソフトウェアの署名をされないハッシュ関数値を有する。

【0069】

この実施形態によれば、ソフトウェアのインスタンスを入手し、ソフトウェアベンダーにより実行されるソフトウェアに名称を割り当て、ソフトウェアのインスタンスに番号を割り当て、第1および第2ハッシュ関数値を算定し、第2ハッシュ関数値に署名し、そしてタグサーバーによって実行されるタグを作るステップがある。

【0070】

本発明は、プロセッサにおいて読み取られ実行される時にソフトウェアのインスタンスを使用するためのリクエストを検出し、ソフトウェアのインスタンスに対応するタグがソフトウェアのインスタンスが使用されることが許容される関連付けられたステータスを持つか否か判断し、タグの真正であることを確認し、タグに対応するソフトウェアのインスタンスが使用管理方式に従って用いられることを確実にするために呼び出し手順を定期的に実行する指示により、符号化されたコンピュータで読み取り可能媒体に関する実施形態を有する。

【0071】

本発明は、通信媒体を通じ搬送波を介して伝送される伝播信号についての実施形態も有する。このような信号は、ソフトウェアのインスタンスに個々に関連付けられた少なくとも1つのタグ、タグテーブルのタグに関連付けられた少なくとも1つの領域、およびタグに関連付けられたソフトウェアの1つのインスタンスに対する使用管理ステータスを示す少なくとも1つの領域を有する符号化されたタグテーブルデータ構造体を搬送する。

【0072】

また、このような信号は符号化された継続メッセージを搬送する。上記継続メッセージは動作に関連付けられたソフトウェアのインスタンスを使用する試みが受信機により検出される時に伝播信号の受信機において実行されるべき動作の指

示を有する。

【0073】

他の方法は、ソフトウェアプログラムが改変されていないことを確かめるためのものである。この方法の実施形態は、ソフトウェアプログラムの内容に関する偽造不能なハッシュ関数値を算定し、偽造不能なハッシュ関数値の結果を前回保持されたハッシュ関数値の結果と比較することにより、結果が同じであるか否か判断し、ソフトウェアプログラムが改変されていれば指示するステップを有する。この方法の1つのバージョンにおいて、オペレーティングシステムが偽造不能なハッシュ関数値を算定し、ソフトウェアプログラムは管理プログラムである。

【0074】

また、本発明によって、データの内容に関して偽造不能なハッシュ関数値を算定し、この値を前に算定されたハッシュ関数値と比較することにより、当該データが改変されていないことを確かめる方法が提供される。好ましくは、管理プログラムは、偽造不能なハッシュ関数値およびこの方法で管理プログラムにより用いられたデータを算定する。

【0075】

上記の実施形態について詳細な記述を行う前に、一般的な高レベルの作用についての要旨を以下に記載して、発明の実施形態の複雑な部分についての理解の助けとする。

【0076】

上記の実施形態に記載されているように、ベンダーの特定のソフトウェアの各インスタンスには、個々の偽造不能なタグが添付される。しかし、同じ特定のソフトウェアのすべてのソフトウェアインスタンスは同一で、暗号化されておらず、それぞれ特定のソフトウェアのコピーからなり、大抵ソフトウェアの名称を有する。例えば、特定のアプリケーションプログラムソフトウェアのインスタンス“スプレッド（展開）”は、スプレッドシートアプリケーションのためのプログラムコードおよび名称スプレッドを有する。本発明では、特他のハードウェア装置が必要でないから、任意の種類のソフトウェアのインスタンスが共通の装置または異なった装置に使用できる。

【0077】

ソフトウェアベンダーは、特定のソフトウェアのインスタンス（コピー）を作り、このソフトウェアの1つのインスタンスを、このソフトウェアのインスタンスについてのタグ番号に対するリクエストと共に、タグサーバーに送る。タグサーバーは、異なったタグのリクエスト番号を作る。各タグは、ベンダーによってソフトウェアの1つのインスタンスで関連付けられ、それが関連付けられたソフトウェアのインスタンスを個々に識別するのに役立つ。ユーザー装置は、ベンダーのソフトウェアのインスタンスを受け取り、使用することを試み、ソフトウェアのインスタンスに個々に関連付けられたタグを確実に受け取る。

【0078】

ユーザー装置は、装置上で作動する管理プログラムを包含し、このプログラムは関連付けられたタグが真正であることを確認し、タグテーブルにタグを格納し、格納装置にソフトウェアのインスタンスを格納し、タグが真正である時にのみソフトウェアインスタンスの使用を許可する。管理プログラムは、インスタンスに関連付けられたタグが真正でない時にのみソフトウェアインスタンスを拒絶する。タグテーブルにおけるすべてのタグは、管理プログラムによりそれに関連付けする“ユーザブル（使用可能）”、“リムーブド（除去された）”または“ペイパーユーズ（使用ごとの支払）”のようなステータスを持つ。管理プログラムは、ソフトウェアのインスタンスを使用するための装置へのコマンドを検出し、ソフトウェアのインスタンスに関連付けられたタグに関連付けられたステータスを確認し、そのインスタンスの使用を許可する。

【0079】

データまたはデータを有するオブジェクトを安全に送りまたは受け取ることは、データまたはオブジェクトに含まれるデータが、承認された送信者または受信者以外の何者かにより改変されまたは開示されることを許さない方法で、送信されまたは受信されていることを意味する。例えば、タグは、ベンダーからユーザー装置へTETS ISPEC、ネットスケープSSL、または通信を確保する他のプロトコルの使用によるネットワークによって安全に送られる。また、タグはタンパー（いたずら）防止シールされた封印に入れられたディスクットを用

いてベンダーからユーザーに渡される。安全な通信は、盗用者による情報の漏洩を防止するために用いられるもので、本発明の保護機構の一部ではない。当事者間の安全な通信のための標準プロトコルがこの目的に用いられる。

【0080】

上記の実施形態に記載されたように、ベンダーソフトウェアのインスタンスに対するタグサーバーにより作られるタグは、そのソフトウェアの名称、ソフトウェアのインスタンスに対する個々の識別番号（以下、インスタンス番号と呼ぶ）、ソフトウェアのインスタンス部分のハッシュ関数値およびすべての前のデータを組み合わせるハッシュ関数値を有する。本発明に用いられているインスタンス番号は、整数または任意のシンボルの連続であることが可能であり、上記の連続は個々の識別子としての役割を果たす。また、タグサーバーは、最後に述べたハッシュ関数値をデジタル的に署名し、タグの中で署名してもよい。

【0081】

以下、署名を有するタグを署名されたタグと呼び、署名を含まないタグを署名されないタグと呼ぶ。ソフトウェアSWのインスタンスINST__SWに対する署名されないタグを作る時には、タグサーバーはソフトウェアSWに関連付けられた、疎で秘密の番号セット（以下、疎セットと呼ぶ）からインスタンスの識別番号を選択する。秘密スパースセットの番号は、例えば物理的处理により生成できる。

【0082】

ソフトウェアのインスタンスINSTに関連付けられたタグが真正であるか否かを判断するために、INSTが設置されまたは使用される装置の管理プログラムは、インスタンス番号INSTのNUM__INSTおよび名称SWのNAME__SWをタグから抽出する。管理プログラムは、ソフトウェアINSTの内容の特定部分に関するハッシュ関数値を算定する。管理プログラムは、次にインスタンス番号NUM__INST、名称NAME__SWおよび前に算定されたハッシュ関数値を組み合わせるハッシュ関数値を算定する。管理プログラムは、算定されたハッシュ関数番号をタグの中で発見されたハッシュ関数値と比較する。管理プログラムは、署名されたタグを構成するデジタル署名を確認する必要がある。署

名のないタグが真正であることは、タグをタグサーバーまたはタグの真正を証明する次述の保護センターに安全に送ることによって、ソフトウェアの関連付けられたインスタンスの最初またはその後の使用を許可する前に管理プログラムによりチェックされる。

【0083】

上述のように上記システムは、タグ付鍵のソフトウェアデータベースおよび確認プログラムを有する保護センターを有する。保護センターは、ユーザー装置にインストールされた各ソフトウェアのインスタンスに対しユーザー装置からすべてのタグを受け取るために、呼び出し手順を介してユーザー装置と定期的に通信する。確認プログラムは、タグが少なくとも1つの使用管理ポリシーに従うことを確保するために、タグ付きのソフトウェアデータベースを用いてユーザー装置から受け取った各タグを検査する。確認プログラムは、ユーザー装置上の各タグに関連付けられたソフトウェアのインスタンスへ試みられたアクセスを続行するための動作を示すユーザー装置へ継続メッセージを戻す。

【0084】

使用管理ポリシーは、少なくとも1つのタグが関連付けられているソフトウェアの個別のインスタンスに関連付けられるか、保護センターが通信する相手の全ユーザー装置に関連付けられるか、または保護センターが通信する相手のユーザー装置の個別のユーザーに関連付けられる。

【0085】

保護センターは、各ユーザー装置の各ソフトウェアのインスタンスに対し各タグに対するタグ付きのソフトウェアデータベースにおけるタグデータ構造体を保持する。各タグデータ構造体は、ソフトウェアのインスタンスのタグ、ソフトウェアのインスタンスの名称、ソフトウェアのインスタンスの固有の番号、ソフトウェアのインスタンスのハッシュ関数値、ソフトウェアのインスタンスに関連付けられたポリシーおよびソフトウェアのインスタンスに関連付けられた一連の呼び出し記録を有する。一連の呼び出し記録の中の各呼び出し記録は、1つの呼び出し手順に関する情報を表し、呼び出し時間、呼び出し手順中に保護センターに搬送されたタグテーブルのヘッダ、前の呼び出し手順のタイムスタンプを示す最

後の呼び出し時間、呼び出し手順中の保護センターに搬送されたタグテーブルのハッシュ関数値および呼び出し手順に関連付けられた継続メッセージに含まれるユーザー装置が続行するための動作を有する。これらの機構を用いることにより、保護センターは、インスタンスの使用ごとに支払うような動作に対しソフトウェアのインスタンスの使用統計を追跡できる。

【0086】

本発明の他の構成によれば、タグのないソフトウェアのインスタンスは、ユーザー装置にインストールすることができる。保護プログラムは、タグのないソフトウェアのインスタンスを検出し、タグのないソフトウェアのインスタンスに指紋処理を実行し、ユーザー装置の指紋テーブルにおける指紋処理から得られた指紋を格納する。保護センターは、この構成によれば指紋データベースを有する。保護センターは、ユーザー装置にインストールされたタグのないソフトウェアのインスタンスに対するユーザー装置からのすべての指紋を受け取るために呼び出し手順を介してユーザー装置と定期的に通信を行う。確認プログラムは、指紋データベースによりユーザー装置から受け取った各指紋を検査することにより、タグのないソフトウェアのインスタンスがソフトウェアの侵害インスタンスではないかを判断する。この方法で本発明は、不正コピーである改変されたソフトウェアの使用を検出できる。

【0087】

確認プログラムがユーザー装置から受け取ったすべての指紋中の1つの指紋と指紋データベース中の指紋との一致を検出した時に、確認プログラムは、報復措置の実行を特定し、確認プログラムは継続メッセージをユーザー装置に戻す。この場合、継続メッセージはユーザー装置に実行されるべき報復措置を示す。従って、ユーザー装置は、例えばタグのない侵害ソフトウェアの使用がつけられた時には機能停止される。

【0088】

また、報復措置は、指紋データベース中の1つの指紋に一致した指紋に関連付けられたタグのないソフトウェアのインスタンスが機能停止されるようにしてもよい。

【0089】

指紋を保護センターで入手するために、ソフトウェアベンダーは、タグのないソフトウェアの侵害インスタンスのコピーを保護センターに送り、保護センターはタグのないソフトウェアの侵害インスタンスのコピーで指紋を算定し、上記指紋を指紋データベースに格納する。

【0090】

本発明の他の実施形態は、装置読み取り可能媒体上で符号化されたタグテーブルデータ構造体を提供する。タグテーブルデータ構造体は、ソフトウェアの1つのインスタンスで固有に識別される少なくとも1つのタグと、タグテーブル中のタグに関連付けられた少なくとも1つの領域を有する。上記の領域は、タグで識別されたソフトウェアの1つのインスタンスに対する使用管理ステータスを示し、タグで識別されたソフトウェアの1つのインスタンスに対する使用統計も示す。タグテーブルデータ構造体は、タグテーブルを固有に識別し、タグテーブルを固有に1つのユーザー装置に関連付けられたタグテーブルヘッダをも有する。タグテーブルヘッダは、ユーザー装置の使用統計に関する情報と、継続メッセージとを有する。継続メッセージは、タグに関連付けられたソフトウェアのインスタンスに対する報復措置および使用管理ステータスを示す。

【0091】

ソフトウェアベンダーは、本発明の1つの構成として、名称とソフトウェアの内容を持つソフトウェアのインスタンスを作るソフトウェア開発機構を有する。各ソフトウェアのインスタンスは、ソフトウェアのインスタンスに固有なタグと結合してのみ実行可能となる。上記のタグは、それが関連付けられ、ソフトウェアの名称、ソフトウェアのインスタンスの固有の番号およびソフトウェアの内容のハッシュ関数値を有するソフトウェアのインスタンスに関する情報の偽造不能な固有な収集物である。ソフトウェアベンダーは、知的所有権を侵害するソフトウェアの侵害インスタンスを検出する侵害ソフトウェア検出機構を有する。ソフトウェアベンダーは、ソフトウェアの侵害インスタンスを保護センターに送ることにより、使用管理はソフトウェアの侵害インスタンスの使用の試みを検出する。

【0092】

本発明の他の実施形態として、テスト中で装置の識別子と一体となったソフトウェアの少なくとも1つのインスタンスを作るソフトウェアベンダーが提供される。上記のテストは、代表的なプログラミング言語の中の“IF文”である。上記テストは、一体となった識別子とソフトウェアインスタンスが使用される装置の識別子とを比較する。一体化した識別子が装置の識別子と等しい時に、ソフトウェアインスタンスは正常に使用できる。等しくなければ装置上の管理プログラムにより報復措置が実行される。さらに、保護手段を追加する為にソフトウェアインスタンスのハッシュ関数値のデジタル署名（一体化した識別子を有する）が送られ、第2テストはデジタル署名が真正であるか否かを判断し、第3テストは署名された値がソフトウェアインスタンスのハッシュ関数値と同じであるか否かを判断する。同じでない時、装置内の管理プログラムにより報復措置が実行される。

【0093】

上記において述べたように、ユーザー装置は、ソフトウェアのインスタンスを受け取り、ソフトウェアのインスタンスに固有に関連付けられたタグを安全に受け取り、そしてソフトウェアのインスタンスにアクセスするようにユーザー装置のユーザーからの試みを受け取る入力部を有する。ユーザー装置のプロセッサは、保護プログラムを実行する。保護プログラムは、ソフトウェアのインスタンスのアクセスの試みを検出し、ユーザー装置のユーザーによるソフトウェアのインスタンスへのアクセスを許可する前にソフトウェアのインスタンスに関連付けられたタグが真正であるかを判断する。保護プログラムは、呼び出し手順が呼び出しポリシーに従っているようにリクエストされているかを判断し、タグテーブルに格納されたタグのステータスを更新するように、呼び出し手順を実行する。呼び出し手順中に、保護プログラムは、タグテーブルをユーザー装置からユーザー装置に結合された相互接続機構を介して送り、タグテーブルの各タグに対して実行される動作を示すユーザー装置へ戻る継続メッセージの受け取りを待つ。この方法で、ユーザー装置は、使用管理ポリシーのセッティングに関係する必要がなく、単にすべての装置に対して中央集散的なポリシーを保持するだけでよい。

【0094】

ユーザー装置にインストールされたタグのないソフトウェアのインスタンスに対し、保護プログラムはタグのないソフトウェアのインスタンスを検出し、タグのないソフトウェアのインスタンスに指紋処理を実行し、ユーザー装置の指紋テーブルにおける指紋処理から得られた指紋を格納する。タグのないソフトウェアに対しては、呼び出し手順中に保護プログラムは、ユーザー装置からの指紋テーブルをユーザー装置に結合された相互接続機構を介して送信し、ユーザー装置に格納されたタグのないソフトウェアのインスタンスに対して実行される動作を示すユーザー装置に戻される継続メッセージの受け取りを待つ。

【0095】

タグのないソフトウェアに対しては、保護センターにおける確認プログラムはタグのないソフトウェアのインスタンスの指紋を受け取るために呼び出し手順を相互接続機構を介して定期的に行う。確認プログラムは、受け取った各指紋を指紋データベースを用いて検査することにより、タグのないソフトウェアのインスタンスがソフトウェアの侵害インスタンスであるか否かを判断し、侵害インスタンスである場合には、確認プログラムはユーザー装置に対して報復措置を生成する。確認プログラムが指紋データベースと受け取った指紋中の1つとの一致を検出した場合には、確認プログラムは実行される報復措置を特定し、確認プログラムは継続メッセージをユーザー装置に送る。継続メッセージは、それを受け取るユーザー装置に行う報復措置を示す。

【0096】

本発明の他の実施形態は、ソフトウェアのインスタンスを受け入れ、ソフトウェアのインスタンスに付き1つのタグの割合で複数のタグを作る真正証明サーバーを提供する。各タグは、関連付けられたソフトウェアのインスタンスを固有に識別し、各タグはタグに関連付けられたソフトウェアのインスタンスの名称に関する符号化された情報、タグに関連付けられたソフトウェアのインスタンスの固有の番号およびタグに関連付けられたソフトウェアのインスタンスで算定されたハッシュ関数値を有する。

【0097】

ソフトウェアへのアクセスを管理する方法において、ソフトウェアのインスタンスを作るステップが実行される。次にソフトウェアのインスタンスに固有に関連付けられたタグが作られる。ソフトウェアのインスタンスおよびタグが次にユーザー装置に配布される、上記の方法は、次にユーザー装置においてソフトウェアのインスタンスへのアクセスの試みを検出し、ソフトウェアのインスタンスへのアクセスの試みが、アクセスされるソフトウェアのインスタンスに関連付けられたタグのステータスを判断することによって有効であるか否かを判断する。

【0098】

タグを作るには、上記の方法は固有の番号をソフトウェアのインスタンスに割り当て、ソフトウェアのインスタンスの内容に関する第1ハッシュ関数値を算定する。第2ハッシュ関数値は、ソフトウェアのインスタンスに対して算定される。第2ハッシュ関数値は、ソフトウェアの名称、ソフトウェアのインスタンスの固有の番号、ソフトウェアのインスタンスの内容および第1ハッシュ関数値を有する。最後に、上記の方法は、ソフトウェアのインスタンスに固有に関連付けられるタグを算定する。タグは、ソフトウェアの名称、ソフトウェアのインスタンスの固有の番号および第2ハッシュ関数値を有する。

【0099】

タグを算定するステップは、署名ハッシュ関数値を作る第2ハッシュ関数値のデジタル鍵署名関数を適用し、タグに署名ハッシュ関数値を有することにより、デジタル的に署名されたタグを作る。これにより、タグの配布を安全にする。公開鍵暗号化技法は、タグをソフトウェアベンダーおよびユーザー装置に安全に配布するのに使用できる。

【0100】

ソフトウェアはユーザー装置のソフトウェアインスタンスを入手し、ユーザー装置におけるソフトウェアインスタンスに関連付けられたタグを安全に入手することによって配布できる。ユーザー装置は、ソフトウェアのインスタンスに関連付けられたタグが署名されているか否かを判断し、署名されている場合にはタグの署名ハッシュ関数値を確認でき、署名ハッシュ関数値が確認されると、ユーザー装置はソフトウェアをインストールすることができる。

【0101】

ユーザー装置におけるソフトウェアのインスタンスへのアクセスの試みを検出するために、本発明の方法は、ソフトウェアのインスタンスへのアクセスに対するユーザーのリクエストを阻止するように、ユーザー装置に保護プログラムを作動させるステップを有する。ソフトウェアのインスタンスへのアクセスへの試みが有効か否かを判断するために、上記の方法は、呼び出しポリシーに基づいて呼び出し手順が必要であるか否かを判断する。上記の方法は、真正であることを確認し、ソフトウェアのインスタンスに関連付けられたタグの使用ポリシーを判断するために、呼び出し手順を実行し、呼び出し手順の結果に基づいてユーザー装置におけるタグ情報を更新する。タグに関連付けられたステータス情報がユーザー装置で検査されることにより、タグに関連付けられたソフトウェアのインスタンスへのアクセスが有効であるか否かを判断する。この方法で、ソフトウェアの保護が行われる。

【0102】

呼び出し手順中に、ソフトウェアのインスタンスに関連付けられたタグを格納するタグテーブルは、ユーザー装置から送られる。上記ユーザー装置はタグテーブルにおける各タグに対して実行される動作を示すユーザー装置に戻される継続メッセージの受け取りを待つ。

【0103】

保護センターは、ソフトウェアのインスタンスに関連付けられたタグを有するタグテーブルを受け取り、タグテーブルに受け取られた各タグをタグ付きのソフトウェアデータベースを用いて検査することにより、タグテーブルの中のタグが少なくとも一つの使用管理ポリシーに一致することを確認する。保護センターは、各タグに関連付けられたソフトウェアのインスタンスへのアクセスの試みを検出した後にユーザー装置が続行する動作を示す継続メッセージを送る。

【0104】

本発明の他の実施形態は、上記の処理に対して指示で符号化された装置読み取り可能媒体を有し、伝播信号は上記の符号化されたタグテーブルデータ構造体を搬送する媒体を通じて搬送波を介して送られる。

【0105】

これらの機構を用いることにより、本発明のシステムは、ソフトウェアのインスタンスにおける正当な権利を有するベンダー／オーナーの権利を保護取り締まる。ベンダーのソフトウェアと実質的に同じであるソフトウェアのインスタンスが密造され、盗まれ、リバースエンジニアリングされ、改変され、または解体されたことを発見することによって、ベンダーに対する権利侵害をベンダーが発見する場合には、システムはソフトウェアの不正コピーの使用を取り締まる。

【0106】

本発明のシステムは、同時にソフトウェアの正当な権利を有するユーザー／オーナーによるソフトウェアについて不正使用の偽りの印象を作り出そうと試みる不正なグループによるサービスの拒絶から、ソフトウェアの正当な権利を有するユーザーを保護する。

【0107】

本発明は、また使用ごとに支払われるソフトウェアに対する各ユーザー装置において、使用ごとの支払い統計値が追跡される。呼び出し手順中、保護センターは、ソフトウェアの使用ごとの支払いインスタンスに対する使用統計値を判断し、費用請求のためにソフトウェアベンダーに使用情報を送る。

【0108】

上述したように、上記システムは、タグ付きソフトウェアデータベースおよび確認プログラムを持つ保護センターを有する。各ユーザー装置は、呼び出し手順を介して保護センターと定期的に通信し、ユーザー装置にインストールされ、または前の呼び出し手順から装置に用いられているベンダーソフトウェアの各インスタンスに対して、そのインスタンスに関連付けられたタグを安全に送る。タグテーブルからの追加データは、完全なタグテーブルを有するまで、呼び出し手順中に保護センターに管理プログラムによって安全に送られる。呼び出し手順は、保護センターまたはユーザー装置により初期化される。保護センターの確認プログラムは、ユーザー装置から受け取る各タグの真正を証明する。

【0109】

本来、確認プログラムは、タグ付鍵のソフトウェアデータベースを用いてユー

ザー装置から受け取った各タグおよび関連付けられたデータを検査することにより各タグの真正を証明し、タグが関連付けられたソフトウェアインスタンスに適用される少なくとも1つの使用管理ポリシーに適合することを確認する。例えば、確認プログラムは、呼び出し中に受け取ったタグが同じ管理プログラムから前の呼び出し時に、コール装置のタグテーブルにおける使用可能な状態、そして同時にその他の装置のタグテーブル使用可能な状態で、使用確認ポリシーの違反の出現があったか否かをチェックする。確認プログラムは、継続メッセージをユーザー装置に安全に戻し、呼び出し手順中に受け取られたタグおよび関連付けられた情報を用いてタグ付きソフトウェアデータベースを更新する。

【0110】

ソフトウェアのインスタンスに対して署名されないタグを作る時、タグサーバーはタグを保護センターに安全に送り、保護センターの確認プログラムは受け取ったタグをタグ付きソフトウェアデータベースに格納する。

【0111】

他の実施においては、タグサーバーはすべての新しく作られたタグを保護センターに送り、保護センターの確認プログラムは受け取った各タグをタグ付きソフトウェアデータベースに格納する。保護センターが呼び出し手順中にユーザー装置からタグを受け取る時に、保護センターの確認プログラムは、タグを保護センターのタグ付きソフトウェアデータベースの中で検索することによって、タグの真正を証明し、発見できない場合には、そのタグが署名されていないタグであって真正でないことを表す。上記のタグが署名されたタグであれば、確認プログラムは、タグをタグ付きのソフトウェアベース中に発見するか、または上記のタグが正しい形状を持ち、タグの中にデジタル署名を有することを確認するかによって、タグの真正を証明する。

【0112】

ユーザー装置への保護センターの継続メッセージは、保護センターにより署名され、タイムスタンプ、タグテーブルのハッシュ関数値、または呼び出し中にユーザー装置の管理プログラムから受け取った他のデータのハッシュ関数値のような識別データを有する。更に継続メッセージは、ユーザー装置の管理プログラム

に対するコマンド（以下、コール動作と呼ぶ）を有する。

【0113】

本発明により用いられる措置例は、以下の事項を含むが、これらに限定されるものではない。

管理プログラムが、（１）特定のソフトウェアインスタンスの継続使用を可能にすること、（２）指定された時間内、ソフトウェアインスタンスの使用を拒絶すること、（３）指定された時間内、所定の名称または所定の指紋リストを有するソフトウェアのインストールまたは使用許可を拒絶すること、または（４）指定された時間内、ユーザー装置の機能を停止することを、指示する。

【0114】

呼び出し手順中に保護センターから継続メッセージを受け取ると、ユーザー装置の管理プログラムが保護センターのデジタル署名をチェックする。管理プログラムは、さらに、継続メッセージが本装置の現呼び出しのためのものであるかを、継続メッセージ中に存在するハッシュ関数値または他のデータと、本装置におけるタグテーブルの一部のハッシュ関数値もしくはタグテーブルのハッシュ関数値またはタグテーブル中に存在する他のデータと比較することによってチェックする。

【0115】

前記署名は認証されたものであることが確認され、前記比較の結果が一致していると認められると、管理プログラムは現呼び出し手順中に、保護センターの回答として継続メッセージを受け入れる。この場合、管理プログラムは、継続メッセージをタグテーブルに格納し、タグの状況を更新し、前記措置に従ったメッセージ中の措置および前記継続報復措置を実行する。

【0116】

使用管理ポリシーは、ソフトウェアインスタンスの個別のタグ、特定のソフトウェアもしくはソフトウェアタイプ、保護センターが通信する全てのユーザー装置、または保護センターが通信するユーザー装置の個別のユーザーに関連付けられることができる。

【0117】

ソフトウェアインスタンスのベンダーによって決められる使用管理ポリシーの例は、以下に示すとおりであるが、これらまたはこれらの組み合わせに限定されるものではない。あるユーザー装置で一旦用いられたソフトウェアインスタンスは、別のユーザー装置では使用されない。あるソフトウェアインスタンスは、同時に2つのユーザー装置で使用されることがないか、または使用可能な状況ではない。あるソフトウェアインスタンスは、特定の装置セットのユーザー装置のみにおいて同時に使用され、または使用可能な状況である。あるソフトウェアインスタンスは、特定回数よりも多く使用されないものとする。あるソフトウェアインスタンスは、特定日付の後には使用されない。あるソフトウェアインスタンスの使用は、そのインスタンスに対するペイパーユーズ（使用ごとに支払う）料金が特定口座に振り込まれた時にのみ、許される。

【0118】

本発明にかかる方法および装置は、ソフトウェアインスタンス、またはソフトウェアインスタンスのクラスの使用に関して、ベンダーまたはベンダー協会によって決められた使用管理ポリシーの遵守を可能にする。

【0119】

保護センターは、ユーザー装置におけるソフトウェアインスタンスに関連した個別のタグに対するタグ付きソフトウェアデータベース中に、タグデータ構造体を保持する。タグに対するタグデータ構造体は、タグ自体に関連付けられ、呼び出し手順中に保護センターにそのタグを転送したユーザー装置に関連付けられるわけではない。各タグデータ構造体は、ソフトウェアインスタンスのタグ、インスタンスがそのコピーであるソフトウェアの名称、ソフトウェアインスタンスのインスタンス番号、ソフトウェアインスタンスもしくはその一部のハッシュ関数値、ソフトウェアインスタンスに関連した使用管理ポリシー、および呼び出し記録の参照の収集物、またはソフトウェアインスタンスに関連した呼び出し記録収集物を備える。前記呼び出し記録収集物の各呼び出し記録は、ある呼び出し手順に関する情報を示す。この呼び出し記録は、また、呼び出し時刻、呼び出し手順中に保護センターに転送されるタグテーブルヘッダのような識別情報、先行の呼び出し手順のタイムスタンプが示す前回呼び出し時刻、呼び出し手順中に保護セ

ンターに伝送されるタグテーブルのハッシュ関数値、および呼び出し手順中にユーザー装置の管理プログラムに送信される継続メッセージを有してもよい。

【0120】

呼び出し手順中に収集されて格納されたデータを用いて、保護センターは、ソフトウェアインスタンスのペイパーユーザに対する請求書を作るために、各ソフトウェアインスタンスの使用統計を編集することができる。

【0121】

タグのないソフトウェアインスタンスが、ユーザー装置でインストールまたは使用されてもよい。管理プログラムは、インスタンスにタグがないことを検出し、タグのないソフトウェアインスタンスの選ばれた部分の指紋を算定し、ユーザー装置の指紋テーブルにこれらの指紋を格納する。この構成にかかる保護センターは、指紋データ構造体を有する。ユーザー装置による上述の呼び出し手順中に、保護センターは、ユーザー装置にインストールされたタグのないソフトウェアインスタンスごとに、ユーザー装置から全ての指紋を受け取る。確認プログラムは、ユーザー装置から受け取った各指紋を指紋データ構造体における指紋と対比することによって、ユーザー装置で用いられているタグのないソフトウェアインスタンスが権利侵害のソフトウェアインスタンスであるかを判定する。この方法によって、本発明は、タグが除去されたベンダーソフトウェアの権利侵害の複製であるソフトウェアインスタンス、またはベンダーソフトウェアの権利侵害から派生した派生物であるソフトウェアインスタンスの使用を検出することができる。

【0122】

保護センターの指紋データ構造体の特定数よりも多い指紋とユーザー装置から受け取った指紋とが一致していることを確認プログラムが検出すると、確認プログラムは、ユーザー装置に戻す継続メッセージに報復措置を記すことができる。このような報復措置の一つによれば、保護センターによってタグのない権利侵害のソフトウェアを用いていることを検出されると、このユーザー装置は指定期間、機能を停止させられる。

【0123】

別の例では、保護センターの指紋データ構造体の指紋に一致した指紋に関連したタグのないソフトウェアインスタンスの機能は停止される、と報復措置が記してもよい。

【0124】

権利侵害ソフトウェアがタグのないソフトウェアとして配布され、または使用されていることを、ソフトウェアベンダーが検知して、このようなタグのないの権利侵害ソフトウェアの複製を保護センターに送ることによって、保護センターの指紋データ構造体は構築される。保護センターは、この権利侵害ソフトウェアの複製の一部についての指紋を算定し、これらの指紋を指紋データ構造体に格納する。

【0125】

ベンダーのソフトウェア権利侵害の防止は、また、タグの有無にかかわらず、ユーザー装置で用いられたソフトウェアインスタンスの選ばれた部分の指紋を作り、これらの指紋をこの装置の指紋テーブルに格納することによっても可能である。上述のように、指紋テーブルの指紋は、呼び出し手順の実行中にユーザー装置の管理プログラムによって保護センターに送られ、保護センターの確認プログラムは、受け取った指紋と保護センターの指紋データ構造体の指紋との間の一致を調べる。本発明のこの構成は、正当なベンダーのソフトウェアの権利侵害バージョンを作ってこの権利侵害ソフトウェアのタグ付きインスタンスを配布する権利侵害ベンダーによって、正当なベンダーの権利が侵害されるのを防止する。

【0126】

装置読み取り可能媒体上の符号化されたタグテーブルデータ構造体を、ユーザー装置はアクセスできる。タグ付きソフトウェアがこの装置にインストールされ、またはこの装置によって使用されると、タグテーブルデータ構造体は、一つのソフトウェアインスタンスに固有に関連した少なくとも一つのタグを有し、タグテーブルのタグに関連した少なくとも一つの領域を有する。この領域は、タグに関連した一つのソフトウェアインスタンスの使用管理状況を示し、タグに関連した一つのソフトウェアインスタンスの使用統計をも示す。タグテーブルデータ構造体は、また、タグテーブルを一意に識別し、このタグテーブルを一つのユーザ

一装置または一つのユーザー装置の管理プログラムに一意に関連付けるタグテーブルヘッダを有してもよい。タグテーブルヘッダは、ユーザー装置の使用統計に関する情報および継続メッセージを有する。継続メッセージは、タグに関連したソフトウェアインスタンスに対する可能な措置および使用管理状況を示す。

【0127】

ソフトウェアベンダーは、名称およびソフトウェアの内容を有するソフトウェアインスタンスを作り出すソフトウェア開発工程を提供する。ベンダーの各ソフトウェアインスタンスは、このソフトウェアインスタンスに関連した固有のタグと共でなければアクセスまたは使用が可能とならない。タグは、関連するソフトウェアインスタンスに関する、偽造が不可能な固有の情報収集物であり、ソフトウェアの名称、ソフトウェアインスタンスの固有の識別番号およびソフトウェアの内容の一部のハッシュ関数値を有する。ソフトウェアベンダーは、また、ベンダーの知的所有権などを侵害するソフトウェアインスタンスを検出する侵害ソフトウェア検出機構を備える。ソフトウェアベンダーが権利侵害のソフトウェアインスタンスの複製を保護センターに送ることによって、権利侵害のソフトウェアインスタンスの使用およびアクセスの企てを検出し、検出すると、このソフトウェアインスタンスを有するユーザー装置に報復措置を課すように、本発明の方法は保護センターによって用いられる。

【0128】

ユーザー装置は、ソフトウェアインスタンスを受け取り、ソフトウェアインスタンスに一意に関連したタグを安全に受け取る入力ポートを有する。ユーザー装置は、また、このソフトウェアインスタンスのインストールまたは使用の要求を受ける。ユーザー装置のプロセッサが管理プログラムを実行する。管理プログラムがソフトウェアインスタンスのインストールまたは使用の試みを検出し、このソフトウェアインスタンスのインストールまたは使用を許可する前に、このソフトウェアインスタンスに関連したタグが認証されたものであること、またはタグに関連した状況を確認する。管理プログラムは、呼び出しポリシーによって決められているように、時々、呼び出し手順が必要であることを決定し、タグテーブルに格納されているタグの状況を更新するために、管理プログラムが呼び出し手

順を実行する。

【0129】

呼び出し手順中、管理プログラムはユーザー装置に結合された相互接続機構を介してユーザー装置からのタグテーブルを安全に送り、ユーザー装置に戻されてくる、タグテーブルのタグごとに実行されるべき措置を示す継続メッセージの受け取りを待つ。この方法により、ユーザー装置は使用管理ポリシーの設定に関して配慮する必要がなく、全ての装置に共通な使用管理ポリシー、またはベンダーによって配布されるソフトウェアインスタンスに関連したベンダーの使用管理ポリシーを遵守するだけでよい。

【0130】

ユーザー装置の管理プログラムによって実施される呼び出しポリシーは、この装置、この装置で用いられる特定のソフトウェアインスタンス、またはこの装置の特定のユーザに関連付けられてもよい。呼び出しポリシーの例は、以下のものを含むが、これらに限定されるわけではない。ユーザー装置の次の呼び出し時刻は、前回の呼び出しからの経過時間、前回の呼び出し以降の装置起動回数および前回の呼び出し以降に装置が使用された全時間の組み合わせによって求められてもよい。同様に、タグ、またはこのタグに関連したソフトウェアインスタンスに関連した呼び出しポリシーが、前回の呼び出しからの経過時間、ソフトウェアインスタンスが使用された回数およびソフトウェアインスタンスがこの装置で用いられた全時間の関数として、次の呼び出し時刻を求めてもよい。ソフトウェアインスタンスに関連した別の呼び出しポリシーは、ユーザー装置においてソフトウェアインスタンスの使用が試みられる度に、呼び出しを実行することを記してもよい。

【0131】

本発明は、呼び出しポリシーによって指定された次の呼び出し時刻以前に、保護センターを呼び出して継続メッセージを保護センターから受け取るのを怠った場合に特定の報復措置を管理プログラムに実行させることによって、前記ユーザー装置または前記ユーザー装置のタグテーブルのタグに適用される呼び出しに適合するように、ユーザー装置およびその管理プログラムを動作させる。前記メッ

セージが実際に前記呼び出し用の継続メッセージとして保護センターによって送られる場合に限り、ユーザー装置の管理プログラムがこの呼び出しに対する保護センターの継続メッセージとして、呼び出し手順の実行中に、受け取ったメッセージを受け入れることを本発明は保証する。これは、上述のように、保護センターがこの継続メッセージに署名し、ユーザー装置の管理プログラムによる現呼び出しに一意にリンクした識別データをこのメッセージに含めること、および保護センターが前記署名および前記識別データを確認することによって達成される。呼び出しポリシーに従った保護センターの呼び出しを行わないことや、不正な継続メッセージを作成もしくは使用する企てによって、ユーザーまたはユーザー装置が本発明の保護の対象から外れてしまうのを、本発明のこの構成要素が防止する。

【0132】

呼び出しポリシーを適合するのに怠った際に前記装置の管理プログラムによって実行されるユーザー装置の前記報復措置の例は、以下の事項を含むが、これらに限定されるわけではない。管理プログラムは、呼び出し手順の実行を除いて、指定期間装置を機能停止させてもよい。ソフトウェアインスタンスに関連した呼び出しポリシーが違反されると、装置がこのソフトウェアインスタンスの使用を指定期間停止してもよい。

【0133】

ユーザー装置でインストールまたは使用される、タグのないソフトウェアインスタンスに対して、管理プログラムは、タグのないソフトウェアインスタンスを検出し、タグのないソフトウェアインスタンスに指紋処理を実行し、この指紋処理から得られた指紋をユーザー装置の指紋テーブルに格納する。タグのないソフトウェアに対して、呼び出し手順中、管理プログラムは、相互接続機構を介してユーザー装置からの指紋テーブルを保護センターに送り、保護センターからユーザー装置への継続メッセージの受け取りを待つ。このメッセージは、ユーザー装置に格納された、タグのないソフトウェアインスタンスごとに、実行されるべき措置を示す。

【0134】

タグのないソフトウェアに対して、ユーザー装置の管理プログラムは、タグのないソフトウェアインスタンスの指紋を相互機構を介して送るために、呼び出し手順を定期的に行う。この呼び出し手順は、ユーザー装置の管理プログラムまたは保護センターによって初期化されてもよい。保護センターの確認プログラムは、受け取った各指紋を保護センターの指紋データ構造体と対比して調べることで、このタグのないソフトウェアインスタンスが権利侵害のソフトウェアインスタンスであるかを判定する。権利侵害のソフトウェアインスタンスであれば、確認プログラムは、ユーザー装置に対する報復措置を準備する。例えば、確認プログラムが、指紋データ構造体における特定ソフトウェアに関連した指紋と、ユーザー装置のタグのないソフトウェアに関連した指紋との間に、十分な数の一致を検出すると、確認プログラムは実行されるべき報復措置を記して継続メッセージをユーザー装置に送信する。継続メッセージを受け取るユーザー装置において実行されるべき報復措置を、継続メッセージは示す。

【0135】

前記タグサーバーは、一般に、特定ソフトウェアのコピーを受け入れ、前記ソフトウェアインスタンスごとに固有のタグを複数生成する。各タグは、関連したソフトウェアインスタンスを一意的に識別する。各タグは、また、タグに関連したソフトウェアインスタンスの名称と、タグに関連したソフトウェアインスタンスの固有番号と、ソフトウェアの前記名称、ソフトウェアインスタンスの前記固有番号、およびタグに関連したソフトウェアの内容を算定したハッシュ関数値を組み合わせたハッシュ関数値とに関する情報を備える。

【0136】

ソフトウェア使用の管理方法において、ソフトウェアインスタンスを作るステップは、上述のように実行される。次に、ソフトウェアインスタンスに一意的に関連したタグが作成される。ソフトウェアインスタンスおよびタグが、次に、ユーザー装置に配布される。次に、ソフトウェアインスタンスのユーザー装置における使用の試みが検出され、使用されるべきソフトウェアインスタンスに関連したタグの状況を判定されることで、ソフトウェアインスタンスのこの使用の試みが、許可されるものであるかが判定される。

【0137】

タグを作るために、本方法は、ソフトウェアインスタンスに固有番号を割り当て、ソフトウェアインスタンスの内容の第1ハッシュ関数値を算定する。次に、本方法は、ソフトウェアの名称、ソフトウェアインスタンスの固有番号、および第1ハッシュ関数値を組み合わせた第2ハッシュ関数値を算定する。最後に、本方法は、ソフトウェアインスタンスに一意に関連したタグを形成する。このタグは、ソフトウェアの名称、ソフトウェアインスタンスの固有番号および前記第2ハッシュ関数値を含む。

【0138】

タグを作るステップは、さらに、タグに含まれる前記第2ハッシュ関数値にデジタル署名関数を適用し、署名されたハッシュ関数値をタグに含めることによって、デジタル的に署名されたタグを生成することができる。

【0139】

ソフトウェアインスタンスおよびソフトウェアインスタンスに関連したタグをユーザー装置に取得させることによって、ソフトウェアが配布されてもよい。ユーザー装置は、ソフトウェアインスタンスに関連したタグに署名されていることを判定し、署名されていれば、タグのハッシュ関数値およびタグにされた署名確認する。この確認が成功すれば、ユーザー装置はソフトウェアインスタンスをインストールまたは使用することができる。

【0140】

ユーザー装置のソフトウェアインスタンスへのアクセスの試みを検出するために、本発明のこの方法は、ソフトウェアインスタンスの使用のユーザー要求を遮るように、ユーザー装置の管理プログラムに訴えるステップを有する。ソフトウェアインスタンスの使用の試みが有効であると判定するために、本方法は、呼び出し手順が呼び出しポリシーに基づいて必要とされているかを判定する。本方法は、呼び出し手順を実行して、認証されたものであることを確かめ、ソフトウェアインスタンスに関連したタグの使用管理ポリシーを決定する。本方法は、また、この呼び出し手順の結果に基づいて、ユーザー装置におけるタグ情報を更新する。タグに関連した状況情報はユーザー装置で調べられて、タグに関連したソフ

トウェアインスタンスの使用が許容されるものであるかが判定される。この方法において、ソフトウェアの使用管理が可能となる。

【0141】

呼び出し手順中、ソフトウェアインスタンスに関連したタグを格納するタグテーブルは、ユーザー装置から保護センターに安全に送られ、ユーザー装置は、タグテーブルの各タグに対して実行されるべき措置を示す、ユーザー装置に戻される継続メッセージの受け取りを待つ。

【0142】

保護センターは、ソフトウェアインスタンスに関連したタグを含むタグテーブルを受け取り、タグテーブルの受け取られた各タグをタグ付きのソフトウェアデータベースに対比して調べ、タグテーブルのタグが少なくとも一つの使用管理ポリシーに従っていることを保証する。保護センターは、各タグに関連したソフトウェアインスタンスの使用の試みを検出すると、ユーザー装置が従うべき措置を示す継続メッセージを送信する。

【0143】

本発明の別の実施形態は、前記処理の指示で符号化されたコンピュータ読み取り可能媒体と、上述のようにタグテーブルデータ構造体を安全に搬送する媒体上を搬送波によって送られる伝搬信号とを有する。

【0144】

これらの機構を用いて、本発明のシステムは、ソフトウェアインスタンスにおける権利の正当なベンダー／オーナーがこれらの権利を警備することを可能にする。ベンダーによって製造されたソフトウェアと基本的に同一動作を行う、密造され、盗まれ、逆行分析され（プログラムを解析され）、または改変されたインスタンスを発見するなどして、ベンダーの権利が侵害されていることをベンダーが発見すると、このシステムは、ソフトウェアのこれら違法コピーの使用を取り締まることができる。

【0145】

本発明のシステムは、同時に、正当なユーザーによるソフトウェアの使用が違法であるという誤った印象を作り出そうとする不正者によって、サービスが拒絶

されてしまうことがないように、ソフトウェアの正当なユーザーを保護する。

【0146】

本発明は、また、使用ごとを基礎として購入されたソフトウェアインスタンスに対する各ユーザー装置におけるペイパーユーズ統計の追跡を可能にする。呼び出し手順中、保護センターは、ソフトウェアインスタンスのペイパーユーズ用の使用統計を求めることができ、料金請求のために使用情報をソフトウェアベンダーに送り返すことができる。

【0147】

【発明の実施の形態】

本発明の前記および他の目的、特徴および利点は、図面に示すように、本発明の好ましい実施形態の説明から明らかであり、前記図面はその全体を通じて同一部品には同一符号が用いられている。図面における寸法は正確ではなく、本発明の原理を示すのに重点が置かれている。

【0148】

図1は、本発明に従って構成された情報システム109の一例を示す。図1には、本発明の主要な構成要素が記載され、本発明に動作上の関係を有するものも記載されている。情報システム109は、複数のユーザー装置104～107と、一つ以上のソフトウェアベンダー101、一つ以上のタグサーバー102および一つ以上の保護センター103とを相互通信する通信ネットワーク100を有する。なお、本実施形態では、ソフトウェアベンダー101、タグサーバー102および保護センター103は、それぞれ、一つだけ示されている。本発明は、ユーザー装置が、ある情報のオーナー、配布者またはベンダーの知的所有権または他の権利を侵害して、その情報をインストールまたは使用するのを防止するように、ユーザー装置104～107の一つによって用いられる情報の使用を管理するもの（図示せず）である。

【0149】

知的所有権または他の権利を保護するために、本発明によって使用が管理される情報は、電子的、磁氣的、光学的などで表現された、いかなるタイプのものであってもよい。情報の例として、コンピュータソフトウェアアプリケーションす

なわちプログラム、データ、ウェブページもしくはウェブサイト、例えばJava（登録商標）アプレットのようなダウンロード可能なアプリケーションプログラム、コンパクトディスク、磁気ディスク、テープ等に記録された電子ブック、画像、ビデオ、音楽などの情報がある。一般に、例えばユーザー装置104～107のようなコンピュータまたは他の装置によって用いられる、あらゆるタイプの情報の使用が管理され、この情報が何であるか、またはこの情報が格納または伝送される実際の物理媒体が何であるかには関係なく、この情報に関する権利は本発明によって保護される。

【0150】

このような情報、および本発明による保護が可能な当業者によって認識されている情報を、以下ではソフトウェアと称する。例えば特定のアプリケーションプログラム、または特定のブックもしくはビデオのコピーのような特定のソフトウェアの個別のコピーを、以下ではソフトウェアのインスタンス、またはソフトウェアインスタンスと称する。ソフトウェアのオーナー、ベンダーまたは配布者を、以下ではベンダーまたはソフトウェアベンダーと称する。装置を使用して、または装置上で、ソフトウェアインスタンスをインストール、使用、実行、読み取り、表示、演奏、鑑賞、印刷、複製、伝送またはアクセスすることを、以下ではそのソフトウェアインスタンスの使用と称する。

【0151】

ユーザー装置104～107は、ソフトウェアの使用に用いられ、これらに限定されるものではないが、コンピュータシステム、ブックリーダー、例えばテーププレーヤ、コンパクトディスクプレーヤ、ミニディスクプレーヤのようなミュージックプレーヤ、ビデオカセットレコーダ、デジタルビデオディスク（DVD）プレーヤ、専用装置などを含むあらゆるタイプの装置である。これらの装置を、以下ではユーザー装置または単に装置と称する。

【0152】

本発明の好ましい実施形態では、ユーザー装置（すなわち104～107のいずれか）はコンピュータシステムであり、前記情報はコンピュータアプリケーションプログラムもしくはデータである。本発明は、このソフトウェアに関するべ

ンダーの権利を保護するように、コンピュータシステムのユーザーによるソフトウェアもしくはデータの使用を管理する機構を提供する。

【0153】

通信ネットワーク100は、本発明の構成要素(101~107)がメッセージまたは信号のような情報を交換するのを可能にする、あらゆるタイプの接続機構である。通信ネットワーク100の例として、インターネット、公衆交換電話網(PSTN)、無線ネットワーク(すなわちセルラーネットワーク)、または他のタイプのコンピュータもしくは情報ネットワークのようなコンピュータネットワークがある。

【0154】

本発明の一般的な運用によれば、一つ以上のソフトウェアベンダーのうちのソフトウェアベンダー100が、ソフトウェアインスタンス(図1には示さず)を作成し、配布する。ソフトウェアインスタンスは、各ユーザー装置104~107で使用されるために、装置104~107でインストールまたは使用される。例えば、ソフトウェアがテープの音楽であれば、テープは図1でテーププレーヤとして図示されているユーザー装置105にインストールされる。ソフトウェアは、ソフトウェアベンダー101から物理的または手動的に搬送され、ユーザー装置104~107に物理的なテープの場合のようにインストールされるか、または公知のデータ伝送機構、すなわちソフトウェアベンダー101からユーザー装置107にソフトウェアのインスタンスをダウンロードする場合のような機構を用いて、ソフトウェアは通信ネットワーク100を介して電子的に伝播されてインストールされる。

【0155】

通信ネットワーク100に結合されたコンピュータシステムであるタグサーバー102は、各ソフトウェアインスタンスに対してタグ(図1には示さず)を作成すなわち生成する。通常、特定のソフトウェアの全てのインスタンスは同一である。好ましくは、一つのタグは、ソフトウェアベンダー101によって作成された一つのソフトウェアインスタンスに一意に関連している。タグサーバー102は、ソフトウェアベンダー101によって作成されたソフトウェアに、好ま

しくは独自の通信経路108を介してアクセスする。タグは、好ましくはソフトウェアの内容、ソフトウェアの名称、および例えばインスタンス番号のようなタグサーバーによって生成された情報やベンダーによって提供された情報などの他の情報に基づいて生成される。タグサーバー102は、また、通信ネットワーク100を用いて、タグを付すためにソフトウェアを取得することもできる。

【0156】

代わりに、各種ソフトウェアの多様なインスタンスを販売する一つのソフトウェアベンダー101が存在してもよい。このソフトウェアベンダー101に対して、一つのタグサーバー102および一つの保護センター103が存在してもよい。タグサーバー102および保護センター103は、ソフトウェアベンダー101の一部であってもよく、すなわち同一コンピュータシステム内に含まれてもよい。代わりに、一つ以上の共有タグサーバー102および共有保護センター103に依存してサービスを受けるソフトウェアベンダー101の協会が存在してもよい。

【0157】

タグがソフトウェアインスタンスに対して作成されると、このタグに対応するソフトウェアインスタンスがインストールされているユーザー装置104~107の一つに、このタグは安全に伝播される。安全なタグの伝播は、好ましくは、例えば安全な通信のためのTETS IPSECやNETSCAPE SSLプロトコルの使用によって、通信ネットワーク100を介して電子的に行われる。手動による安全なタグの伝播も、本発明のシステムによって用いられる。手動による安全なタグ伝播の一例として、タグおよび可能ならば関連したソフトウェアインスタンスを含む、改変されないパッケージ内でタグを配布する。

【0158】

ソフトウェアのインスタンスおよびソフトウェアインスタンスに関連したタグがユーザー装置104~107に一旦インストールされると、この装置のユーザー（図示せず）またはこの装置自体は、このソフトウェアの使用を試みることができる。しかし、ソフトウェアインスタンスの使用が許可される前に、ユーザーまたはソフトウェアを有するユーザー装置104~107によって要求されたソ

フトウェアインスタンスに対する有効なタグがこの装置内に存在することを、このユーザー装置の管理プログラム（図示せず）が確認する。定期的に、各ユーザー装置が保護センター103と通信ネットワーク100を介して通信する。これにより、ユーザー装置上のソフトウェアインスタンスに関連した全てのタグが、有効であることおよび使用管理ポリシーに従って用いられていることが保証される。

【0159】

すなわち、本発明は、ソフトウェアインスタンスの装置による使用が、有効な関連タグの存在にリンクされていることを保証する。タグは、ユーザー装置を保護センターに通信させることによって、使用特性が定期的に確認およびチェックされる。遵守される使用管理ポリシーの例では、タグは一つの装置にのみ存在する。ユーザー装置104～107がソフトウェアインスタンスを使用することができるか否かの判定は、ユーザー装置と保護センター103との間で実行される後述の呼び出しと呼ばれるタグ処理手順に基づく。

【0160】

本発明にかかる実施形態のさらに詳細な説明を行う前に、以下の表1（表1A～表1Cから構成される）に、本発明に関連する各種の要素の理解を助けるための用語集を示す。

【0161】

【表1A】

表1A：用語の定義（その1）

用語	定義
ACTIONS(措置)	装置で使用してもよいソフトウェアを記載し、検出されたベンダーソフトウェアの不正使用に対する報復措置を記す継続メッセージCMに含まれる措置指令。
CALL-UP_POLICY_SW (呼び出しポリシー)	特定のソフトウェアSWまたは特定のソフトウェアインスタンスINST_SWに関連付けられ、任意に指定された呼び出しポリシーであって、装置が何時保護センターに呼び出し手順を実行しなければならないかを規定する。
CM(継続メッセージ)	保護センターからユーザー装置に送られる継続メッセージであって、ユーザー装置のソフトウェアインスタンスの使用許可状況を示す。
DEVICE IDENTIFIER (装置識別子)	ハードウェア識別子から、または管理識別子ID(SP)を用いることによって、装置を識別する方法。この識別は、各ソフトウェアインスタンスがテストにおいて装置識別子を組み込む実施形態で用いられる。
FP(X) (指紋)	入力ストリングXに指紋関数（例えばハッシュ関数）を施して算定される指紋。
GC	保護センター。
HASH_INST_SW	HASH_SW, NAME, NUM_INST_SWおよび他の領域から算定されるハッシュ関数値。
HASH_SW	ソフトウェアSWの内容から算定されたハッシュ関数値。各ソフトウェアインスタンスSWは同一のHASH_SWを有する。HASH_SWはHASH(SW)の他の表記である。HASH_SWは、ソフトウェアの一部のみのハッシュ関数値結果の場合もある。
ID(X), ID(SP)	オブジェクトXに任意に関連付けられた固有の識別番号。例えば、ID（管理プログラム）は、起動イベントが起きた時点と、保護センターによって提供される情報および一つ以上のメモリ位置の値を含む他の情報とを組み合わせることによって、最初に装置が起動した際に算定された管理プログラムの識別番号である。

【0162】

【表1B】

表1 B : 用語の定義 (その2)

INF_SW (権利侵害のソフトウェア)	ベンダーによって確立された知的所有権または他の権利を侵害する、ベンダーのソフトウェアSWの認証されていない複製または派生物。権利侵害のソフトウェアの配布を検出し、このソフトウェアの権利侵害的な使用を防止する法律上の権利を、ベンダーが持つこととする。権利侵害のソフトウェアには、タグが不適切な方法で除去されたもの、タグが変更されたもの、または装置識別テストがある場合にこれが変更されたものが含まれる。
INST_SW (インスタンス)	ソフトウェアSWの全インスタンスセットから選ばれた特定のソフトウェアの特定のインスタンス (複製またはコピー)。ソフトウェアSWの全てのインスタンスは同一である。
NAME_SW	特定のソフトウェアSWの名称。
NUM_INST_SW (インスタンス番号)	特定のソフトウェアインスタンスINST_SWに関連付けられた固有番号。この番号は、数字、アルファベット、文字もしくは記号が混ざった列、または他のパターンである。同一の原則が上記識別ID(X)に適用される。
POLICY(TAG_INST_SW) または USAGE SUPERVISION POLICY (使用管理ポリシー)	知的所有権およびアクセス権の保護、またはソフトウェアに関連付けられたペイパービュー (観るごとに支払う) 使用制限に関するソフトウェアベンダーまたは他の組織によって規定されたポリシーおよび規制。これらポリシーおよび規制は、特別のソフトウェアインスタンスに依存してもよい。POLICY(TAG_INST_SW)は、保護センターGCおよび管理プログラムSPによって遵守される。
SP, SUPERVISING PROGRAM	管理プログラム。ユーザー装置に一体化されたプログラムであって、本明細書に記載されている、ユーザー装置のソフトウェアインスタンスの使用管理を行うメカニズムを提供する。
PRIVATE_KEY_X	デジタル的な署名を生成するためのXによって用いられる秘密鍵。
PUBLIC_KEY_X	署名をチェックして認証するために、Xによってデジタル的に署名されることが意図された、データの受領者によって用いられる公開鍵。
SIGN_TS (署名)	タグサーバのデジタル署名。

【0163】

【表1 C】

表1C：用語の定義（その3）

SIGN_X(M)	メッセージMのXによるデジタル的な署名であって、以下の特性を有する。(1)XのみがSIGN_X(M)を生成することができる。(2)デジタル的な署名の受領者は、XがMを署名していることを確認できる。
SPARSE_SET (疎セット)	疎で秘密の番号セットであって、一実施形態においては、この番号セットから固有のインスタンス番号が全てのソフトウェアインスタンスに選ばれる。インスタンス番号は物理的処理によって生成されてもよい。
SPARSE_SET_SW (疎セット)	疎で秘密の番号セットであって、一実施形態においては、この番号セットから固有のインスタンス番号NUM_INST_SWが一つの特定のソフトウェアSWのインスタンスに選ばれる。これにより、ソフトウェアXのインスタンスはソフトウェアYのインスタンスと同一のインスタンス番号を持つことができる。インスタンス番号は物理的処理によって生成されてもよい。
SW (ソフトウェア)	本発明によって保護される特定のベンダーソフトウェアである。例えばスプレッド(Spread)という名称のソフトウェアコードがある。
TAG_INST_SW	偽造が不可能な固有の署名がされているか、または署名されていないタグであって、特定のソフトウェアインスタンスINST_SWに関連付けられている。
TAG TABLE (タグテーブル)	ソフトウェアインスタンスに関連付けられたタグに関する情報と、装置のソフトウェアインスタンスの使用または使用管理に関する情報とを含む装置に格納されたテーブルまたはファイル。
UNTAGGED_SW (タグのないソフトウェア)	関連付けられたタグTAG_SWを持たず、ユーザーがユーザー装置でインストールまたは使用することを試みるソフトウェア。例えば、シェアウェアもしくはフリーウェアまたはユーザーによって作成されたソフトウェアである。
VRP (確認プログラム)	保護センターGCにおける確認プログラム。

【0164】

技術用語の詳細な定義：

本発明の一定の実施形態は、本質的に複雑である。したがって、本発明の一定の実施形態によって用いられる技術用語に対する、他の補助的な定義を以下に示す。

1. 指紋関数またはハッシュ関数F：XとYが等しくなければ、F(X)とF(Y)が等しくない可能性が高くなるように、データXを、より小さいデータF(X)に対応づけるで数学的関数。ハッシュ関数の例として、Xはバイト列であ

る。さらに、好ましくは、数値 p はランダムに選ばれるが、以後は固定された 64 ビットの素数である。バイト列 X は、ある数値 (256 ベースで書かれ、バイトはその数値の数字である) および $F(X) = X \bmod p$ (X を p で割った余り) と見なされる。これにより、 $F(X)$ の値は、 X がいかに大きくても 64 ビットストリングである。

2. 偽造できないハッシュ関数 H : 所定の X に、 $H(X)$ を算定するのは容易であるが、 $H(X) = H(X')$ かつ X と X' は異なるような、 X' を生成するのが困難な性質を有する指紋関数。用語「困難」は、現在の技術水準によれば X のサイズでは必要な演算時間が一般に指数関数的であるか、または事実上実現不能であると一般に理解されることを意味する。偽造できないハッシュ関数の例として、MD5 がある。

【0165】

3. ソフトウェアインスタンスの使用 : 装置の使用によって、または装置において、ソフトウェアインスタンスの、インストール、使用、実行、ラン、接続、読み取り、記憶媒体の検索、または記憶媒体の内容変更、表示、演奏、鑑賞、印刷、複製、伝送またはアクセス。

4. ソフトウェアインスタンス部分は、ソフトウェアインスタンスのテキストもしくはデータの全て、またはテキストもしくはデータの一部のシーケンスを含む。

5. 指紋処理 : データアレーの位置シーケンスが与えられた際における、これらの位置の値に対する関数値の算定。例えば、位置 16, 32 および 64 が、それぞれ、値 3, 4 および 17 を持つ場合、指紋処理は、3, 4 および 17 の関数値を算定する。この関数は、単にこれらの値のリスト (この例では 3 つの数値)、またはこれらの値のリストのハッシュ関数である。別の例では、位置は、 $i_1 \sim j_1$, $i_2 \sim j_2$, \dots , $i_k \sim j_k$ であってもよい。指紋処理は、アレーのこれら k シーケンスごとにハッシュ関数値を算定し、 k 個の算定値をリストに挙げる。

【0166】

6. 指紋検査 : 2 つの指紋シーケンスを比較する方法。本発明は、2 種類の指

紋検査を用いる。2種類の検査は、同一位置指紋検査と一般位置指紋検査である。両方の指紋検査において、指紋リストは、位置リストの一つのリストにおける値に基づいて算定される。例えば、リストに3つの指紋 f_1 、 f_2 および f_3 があり、 f_1 は位置 10、20、30 および 40 の値から算定され、 f_2 は位置 30 および 60 の値から算定され、 f_3 は位置 100 および 200 の値から算定されるとする。このリストを送信リスト (Send List) と称することとする。両方の指紋検査において送信リストの受信者は、送信者と同一の指紋リスト位置における値に基づいて指紋リストを算定する。この指紋リストは、受信リスト (Receive List) と呼ばれる。

同一位置指紋検査では、送信リストの各要素が受信リストの該当する要素に等しい場合に、一致したと宣言される。具体的には、送信リストの第1要素が受信リストの第1要素に等しく、送信リストの第2要素が、受信リストの第2要素に等しく、他の要素についても同様となる。

一般位置指紋検査では、位置に関係無く、送信リストおよび受信リストに多数の共通要素がある場合に、一致したと宣言される。この多数は、ポリシーおよび指紋が作られるデータテキストの長さに依存する。この長さはパラメータ k と定義される。 k が例えば 50 バイトであれば、保護センターの指紋データ構造体 (図9の符号 137) のリストと同一のソフトウェアを装置リスト (Device List) があらわすためには、一つまたは少数の一致で十分である。さらに、一定の一致には他の一致よりも重点が置かれ、重点が置かれていれば少ない一致でも十分である。

指紋の送信リストを送るのに加えて、送信者は、送信リストを生成した値を有する位置リストのリストを送ってもよい。これにより、指紋の計算は、予測し得ないランダム処理に依存する。

【0167】

7. 偽造不可能：ベンダーの要求に基づいてタグを生成するタグサーバー 102 (図1) によって用いられる秘密情報を知ることなしに、有効なタグを生成するのは、敵対者にとって計算上不可能な場合、タグは偽造不能である。本発明では、デジタル署名 (図3A) および疎セット (図3Bおよび3C) を、タグの偽

造不可能を達成する2つの好ましい方法として用いる。

8. 安全な伝送：Xが転送されるネットワークプロトコルを観察すること、またはXを転送するパッケージを見ることを他者もできるが、意図された受取人のみがXを見ることができるような値Xを送る方法。信頼できる宅配業者によって配達される封印された包みは、包みの中身を安全に送る一方法である。安全な通信のためのT E T S I P S E CやN E T S C A P E S S Lプロトコルの使用によってメッセージを送ることは、通信ネットワーク100（図1）上の安全な伝送を保証する別の方法である。

9. 事象履歴：全ての使用の試み、成功した使用、使用期間、および／またはタグテーブルに関連した起動のようなその他の事象の経時的な記録である。2つの装置が、たとえ同一のソフトウェアインスタンスおよび同一の識別子を有していても、同一事象履歴を有することはありそうもない。事象履歴は、一人以上のユーザーよる特定の装置の使用の経時的な記録に基づいてもよい。

【0168】

図を用いた説明に戻る。図2には、本発明に従って構成されたシステム109のアーキテクチャーの詳細を示す。図2を、本発明の全動作の総合的な記載の概説に用いる。この説明全体を通して、本発明の各構成のより詳細な部分の記載には、他の図が参照される。

【0169】

システム109の動作において、ソフトウェア（SW1、SW2、SW3、SW4とされている）のインスタンス（INST_SW）111～114は、ソフトウェアベンダー101によって作成され、ベンダーストレージ110（記憶装置）に格納される。一つ以上のソフトウェアベンダー110が存在してもよい。ソフトウェアベンダー101の例としては、出版社（複製可能なパフォーマンスの記録、または電子的に読み取り可能な本を作成）、コンピュータソフトウェア開発者（コンピュータソフトウェアアプリケーションプログラムを作成）、データ収集会社（情報データベースを作成）、個人プログラマ等がある。ソフトウェアベンダー101によって生成されるソフトウェア（SW）は、情報、データまたはコードを含む実際のソフトウェアの内容（SW）を意味する。ソフトウェア

(SW)は、関連した名称(NAME__SW)を有してもよく、この名称は通常ソフトウェアベンダー101によって割り当てられる。ソフトウェアの各インスタンス(INST__SW)111~114は、名称を持つソフトウェア(SW)の別個の物理的な複製(コピー)として考えられる。すなわち、特定のソフトウェア(SW)に対する各ソフトウェアインスタンス(INST__SW)は、同一の名称(NAME__SW)および同一のコード、データまたは他の情報内容を持つソフトウェア(SW)のコピーに過ぎない。

【0170】

例えば、文書作成アプリケーションプログラムがソフトウェアベンダー101によって作成され、名称(NAME__SW)に「ライト(W r i t e)」が与えられるならば、ライトプログラムを備えたバイナリまたは実行可能なコード、データまたは他の情報が、ソフトウェア(SW)と呼ばれる。ライトソフトウェア(SW)の個別のコピー(例えば、プログラムのコピーを含む各ディスク)はこのソフトウェアの別個のインスタンス(INST__SW)であるが、同一のソフトウェア内容(SW)を持つ。したがって、図2において、各インスタンス111~114は同一のソフトウェア内容(SW)を含んで各インスタンス111~114は同一の名称(NAME__SW)を持つか、または各インスタンス111~114は異なったソフトウェア(SW)(すなわち異なったデータ、コードまたは他の情報)のコピーを示し、異なったソフトウェア内容(SW)を持つ各インスタンスの名称(NAME__SW)111~114は通常異なる。

【0171】

タグサーバー(TS)102は、ベンダー101の要求に応じて、各ソフトウェアのインスタンス111~114用の偽造が不可能な固有のタグ(TAG__INST__SW)120を作成する。本発明の好ましい実施形態では、一つの固有のタグは、一つのソフトウェアインスタンスに対して準備され、このインスタンスに関連付けられる。別の実施形態では、複数の固有のタグが一つのソフトウェアインスタンスに関連付けられることはあるが、好ましくは、2つの異なるソフトウェアインスタンスが共通の関連タグを共有することはない。

【0172】

要求されたタグを作成するために、TS102（図1）は、作成しようとするタグのインスタンスに対する各特定ソフトウェアのコピーを取得する（図3A、3Bおよび3Cのステップ150）。例えば、TSは、「ライト7.2」の一つのコピーを有してもよい。ここで、「ライト7.2」は、プログラムファミリーライトのリリースすなわちバージョンである。一般に、タグ102は、特定のソフトウェアインスタンス（INST_SW）（すなわち111～114の一つ）に関連した、偽造が不可能で固有のデータビットシーケンスである。後述するように、本発明の実施形態によれば、ユーザー装置104は、ソフトウェアインスタンス111～114に関連した有効なタグ120を最初に調べることなしに、ソフトウェアインスタンス111～114を使用することはできない。

【0173】

ソフトウェアインスタンス111～114用のタグ120は、好ましくは、ストレージ装置200のタグテーブル210に格納される。ストレージ装置200は、ユーザー装置104に結合されているか、またはユーザー装置104と一体となっている。ソフトウェアインスタンス111～114は、タグテーブル210に格納されているソフトウェアインスタンス（111～114の一つ）に関連したタグ120を引用し、このタグ120が使用状況を持つ場合にのみ、ユーザー装置104で用いられることができる。タグの状況としては、例えば図6のタグテーブルの第2列に、「使用状況」が示されている。すなわち、一定のソフトウェアは、このソフトウェアのインスタンスにタグが存在する時にのみ、ランできる、という表示を有する。（侵害者は、この表示を除去するかもしれないが、この場合、後述するタグのないソフトウェアに対する保護機構が適用される。）このようにして、本発明の構成は、ユーザー装置104に存在するソフトウェアインスタンスに特に関連した有効なタグを要求することによって、一定の実施形態におけるソフトウェアの使用の権限を許可して与える。

【0174】

さらに後述するように、タグの作成、タグの確認およびタグの実施を追跡および管理するために、本発明に従って構成されたシステムの構成部分の能力は、ソフトウェア使用権限に対して、従来のシステムよりも固有の利点を提供する。図

2のシステム109のまだ説明をしていない構成部分について説明する前に、タグ作成について詳述する。

【0175】

図3A、3Bおよび3Cは、本発明に従って構成されたタグサーバー102におけるタグ作成処理中に行われる処理ステップの好ましい実施形態を示すフローチャートである。これらの図は、互いに似ているので、ステップ番号の多くは同一であり、以下これらの図について同時に説明する。

【0176】

ステップ150において、タグサーバー102は、ローカルストレージから名称を持つソフトウェア（NAME__SW、SW）のコピー111～114を取得する。このソフトウェアは、タグが付されるものである。さらに、タグサーバー102は、ベンダー101からタグ要求（図2）を取得する。ステップ151A（図3A）、151B（図3B）および151C（図3C）において、タグサーバー102は、固有番号（NUM__INST__SW）を生成する。図3Aのステップ151Aでは、番号は単に固有である。しかし、図3Bのステップ151Bおよび図3Cの151Cでは、固有番号（NUM__INST__SW）は疎セット118（図2）から選ばれる。

【0177】

疎セット118（図2）は秘密番号のセットであって、この秘密番号のセットから名称を持つソフトウェア（NAME__SW、SW）のインスタンスそれぞれに対してインスタンス番号（NUM__INST__SW）が選ばれる。好ましくは、利用可能な番号の範囲（例えば、特定のソフトウェアに1億のインスタンスが存在し、64ビットにより定められる範囲においては100億以上の可能な番号が存在する場合）に比較して、前記番号の数は比較的少ない。したがって、セット118は、疎（sparse：散在する、まばらな）と称される。

【0178】

疎であることより、敵対者または侵害者にとって、有効なインスタンス番号を生成するのは困難である。全てのソフトウェアに対して一つの疎セットが存在してもよい。または、関連したインスタンスのセットにより決められた各特定のソ

フトウェアに対して、それぞれ異なる疎セットが存在してもよい。好ましい実施形態では、一つの疎セット118が、全てのソフトウェアに対するインスタンス番号の源として用いられる。しかし、各特定のソフトウェアに対して別の疎セット118を持つことによって、より単純なインスタンス番号生成の配布管理を可能にしてもよい。

【0179】

例えば、上述の「ライト」アプリケーションソフトウェアに関連した疎セット番号118 (SPARSE__SET__SW) が存在してもよい。この疎セット番号118から、ライトソフトウェアの各インスタンス (INST__SW) に対してインスタンス番号 (NUM__INST__SW) が選ばれる。安全上の理由から、疎セットの新しい番号は、要求があり次第、例えば光電計数装置 (本発明では図示せず) のような物理的処理へのアクセスによって、実現または生成されてもよい。

【0180】

ステップ152 (図3Aおよび3B) において、タグサーバー102は、ソフトウェア (SW) の内容またはその内容の一部についてハッシュ関数値を算定する。好ましい実施形態において、同一ソフトウェア内容SWを有する2つ以上のソフトウェアインスタンス (INST__SW) 111~114にタグが付されている場合、ハッシュ関数値HASH__SWがソフトウェア (SW) に対して1回だけ算定される。これは、各インスタンス111~114が同一のコード、情報および/またはデータを含む、すなわち同一のSW内容を持つからである。さらに、値HASH__SWは、全てのソフトウェアのコピーごとに検索または生成されるよりも、タグサーバー102によって1回検索または生成されるだけが必要とする。本発明のこの構成は、同一ソフトウェア (SW) の多くのインスタンスにタグを付す際には、タグ作成時間を節約する。このような場合、ハッシュ関数値HASH__SWの算定の必要性は1回限りである。別の実施形態では、ソフトウェア内容の一部のみについてのハッシュ関数値の算定によって、さらに最適化される。これは、タグサーバー102とユーザー装置104~107の両者におけるハッシュ関数値の確立に必要な時間が、さらに減少するからである。

【0181】

ステップ153（図3A、3Bおよび3C）においてソフトウェアインスタンス（INST_SW）に関連したタグに組み込まれる第2ハッシュ関数値HASH_INST_SWが算定される。ステップ153は、ステップ152とは異なる。ステップ152において算定されたハッシュ関数値HASH_SWは同一ソフトウェアSWのすべてのインスタンスINST_SWに対して同一であるのに対し、ステップ153においてはハッシュ関数値HASH_INST_SWが同一ソフトウェアSWの各NUM_INST_SWに対して固有である。一実施形態では、第2ハッシュ関数値HASH_INST_SWは、ソフトウェアの名称（NAME_SW）、ソフトウェアインスタンスの固有番号（NUM_INST_SW）およびステップ152で既に算定されたハッシュ関数値HASH_SWを共に組み合わせる。名称およびソフトウェアのみ、またはソフトウェアおよび番号のみ、などのような他の組み合わせは、当業者には理解される同様の機能を提供するものとして認識される。このようなハッシュ関数を介してエンコードされたデータの組み合わせは、本発明の範囲内に含まれる。

【0182】

ハッシュ値HASH_INST_SWがソフトウェア111～114の各インスタンスに対して算定された後に、署名されたタグ（図3A）または署名されていないタグ（図3Bおよび3C）が、ステップ154Aおよび154Bによって、これらのインスタンス111～114に対して作成される。図3Aのステップ154Aにおいて署名されたタグがソフトウェアインスタンス111～114に対して作成されるのに対し、図3Bおよび3Cのステップ154Bにおいては、署名されていないタグがソフトウェアインスタンス111～114に対して作成される。準備されたタグのデジタル署名のある部分によって、たとえばインスタンス番号が、例えば続き番号であるために予測できる場合であっても、偽造不可能であることを、署名されたタグが保証する。署名されていないタグは、この保証機能を提供することはできないが、ステップ154Bにおいて生成された署名されていないタグは、好ましくは疎セット151Bから選ばれたインスタンス番号NUM_INST_SWを含むので、この代わりの方法によって、タグの偽造不

可能が保証される。署名されたタグTAG__INST__SWは、ステップ154 Aにおいて、以下のように算定される。

$$\text{TAG_INST_SW} = (\text{NAME_SW}, \text{NUM_INST_SW}, \text{HASH_INST_SW}, \\ \text{SIGN_TS}(\text{HASH_INST_SW}))$$

ここで、SIGN__TSは、HASH__INST__SWハッシュ関数値において実行されるデジタル署名関数である。デジタル署名SIGN__TSは、全ての考えられる敵対者および図2のタグサーバー102自体を除いたあらゆる団体（ユーザー装置、ベンダー等）から秘密に保たれているデジタル的な鍵（キー）である秘密鍵（プライベートキー）PRIVATE__KEY__TS117を用いて、タグサーバー102によって生成される。

【0183】

署名されていないタグTAG__INST__SWは、ステップ154 B（図3 B）において以下のように算定される。

$$\text{TAG_INST_SW} = (\text{NAME_SW}, \text{NUM_INST_SW}, \text{HASH_INST_SW})$$

【0184】

タグサーバー102によってタグTAG__INST__SWが生成された後、タグは、好ましくは、図2のタグ120によって示され、図3 Aおよび3 Bのステップ156に関して後述するように、タグを要求するソフトウェアベンダー101および保護センター103に安全に送られる。保護センター103では、図9の符号129、138に関して後述するように、タグ120は各種のタグデータベースに格納される。

【0185】

ソフトウェアインスタンス（例えば111）に関連したタグ120、およびタグ120がタグサーバー102によって準備される方法は、本発明の多数の重要な目的に合致する。

(1) 関連した有効タグ120、好ましくは装置（例えば104）のタグテーブル210（図6に詳細に示されている）に保持されているタグ120を装置104が格納またはアクセスしない限り、かつ、このタグ120が、関連インスタンス111の適切な使用を許可または指示するタグテーブル210に使用状況（

図6の第2列)を持たない限り、装置104はベンダー101のソフトウェアインスタンス111を使用することはできない。

(2) 装置(例えば104)と保護センター103の間の後述する命令された呼び出し手順(図12、13Aおよび13B)によって、保護センター103は、タグ属性を管理、認証、追跡、確認および一般的に監督し、タグ120に関連したソフトウェアインスタンス111が、ソフトウェアインスタンス111に対するベンダー101の使用管理ポリシーに従って用いられていることを保証する。この使用管理ポリシーは、好ましくは保護センター103によって保持されている。

(3) タグ120の偽造が不可能で、タグ120が好ましくは安全な方法で送られることより、ベンダー101(またはタグサーバー102)からタグ120を合法的に取得し、このソフトウェアインスタンス111に対するベンダー101の規定された使用管理ポリシー(この図には示さず)に従って関連したソフトウェアインスタンス111~114を使用するユーザーまたはユーザー装置104のみが、このタグ120を持つことについて保証される。本発明のこの構成は、敵対者または侵害者が有効なタグ120のコピーの生成および/または使用を試みるのを防止する。これにより、本発明の機構によって、コピーする敵対者/侵害者にも、ソフトウェアインスタンス111および関連したタグ120を用いる合法的なユーザーまたはユーザー装置に対するのと同様に、報復措置を作り出す。

【0186】

タグ120には、いくつかの別の構成が考えられる。別の構成の一つでは、以下に示すように、フィールドのサブセットを持つ。特に、ハッシュ値HASH__INST__SWはタグ120に含めず、NAME__SWおよびNUM__INST__SWをタグ120に残す。このような実施形態の利点は、システム構成要素(例えば101、102、103、104)の間で送られ、各タグ120について算定されるデータが少ないことである。不利な点は、タグ120のオーナーが、異なる特定のソフトウェアインスタンス111にタグ120を関連させることを試みるかも知れないことである。これは、値HASH__INST__SWがHAS

H__SWに依存し、インスタンス111内のソフトウェアSWが正しいか、または変更されていないかを確認するために、HASH__SWが用いられるので、HASH__INST__SWがタグ120で利用される際に防止される。

【0187】

第2の別のタグ構成は、NAME__SW、NUM__INST__SW、HASH__SWである。この構成を用いることにより、全てのタグ120は、その内容（すなわちSW）がHASH__SWに対するハッシュ関数に一致するソフトウェアに関連される。この方法の不利な点は、正しく見える違法タグ120を、侵害者が生成できる可能性があることである。ソフトウェアの使用を保護するために選ばれる本発明の実施形態の複雑性に応じて、本明細書に記載のシステムは、各種の記載した問題を解消するように構成されている。

【0188】

別の例として、タグ120の第3の別の構成は、NAME__SW、NUM__INST__SW、HASH__SW、SIGN__TS（NAME__SW、NUM__INST__SW、HASH__SW）である。このタイプのタグ120では、デジタル署名がタグの偽造を防止する。これは、好ましくはタグサーバー102のみが署名関数SIGN__TSの演算に必要な秘密鍵SECRET__KEY__TSを持つからである。

【0189】

取り除かれてもよい別のタグ領域は、領域NAME__SWである。この実施形態の利点は、システム構成要素間で送られるデータの量を減らすことである。タグがINST__SWをランまたは使用するのに存在しなければならない名称以外の手段によって、ソフトウェアインスタンスINST__SWを示すのであれば、名称は不必要となる。名称のないタグは、例えば所定のソフトウェアベンダー101から配布されるソフトウェアが1種類しか存在しない場合、ソフトウェアベンダー101の識別子は、このベンダーによって生成されたソフトウェアに対する名称として用いられることができる。代わりに、NUM__INST__SWが、全ての種類のソフトウェアを通して固有である場合、NAME__SWは不要である。

【0190】

タグ120から取り除かれてもよい別の領域は、NUM__INST__SWである。このタグ構成の利点は、ネットワーク100上を送られるデータの量が減少し、固有の番号選択処理（例えば図3A、3Bおよび3Cに説明されているステップ151）を必要とすることなく、より簡単なタグ作成方法を用いることができることにある。不利な点は、同一のNAME__SW（この領域が保持されているならば）を持つ、異なるタグの識別ができなくなって、二重のインスタンス111～114が許容されてしまうことである。

【0191】

タグの別構成の他の実施形態は、任意の領域Aを含む。ID（SP）（図4の209-A）と表されたユーザー装置（例えば104）の管理プログラム（後述する図4の符号209）の固有識別子が、例えばハードウェア特定装置、可能ならば装置104の管理プログラム209が初めて起動された時点、および可能ならば保護センター103から装置の管理プログラム209によって安全に取得された固有番号と、装置内の少なくとも一つのメモリ位置の値との組み合わせから算定される。これは後に詳述するが、ここでは各種のタグ作成処理の理解のために記述する。ユーザー装置104～107で用いられるソフトウェアインスタンス111に関連したタグ120にユーザー装置104～107の管理プログラム209の識別子ID（SP）209-Aを含めることで、以下に詳述するように、割安な保護センター103呼び出しをサポートすることができる。

【0192】

本発明の別のタグおよびタグ作成の実施形態に含まれる任意の領域は、ソフトウェアインスタンスINST__SW内データの特定の位置における指紋リストである。指紋について後述するが、その名称が示唆するように、指紋はソフトウェアインスタンスから選ばれる一部分もしくは複数部分またはデータ領域の固有のエンコードされたものである。指紋の使用方法は、図3Cのステップ151Dおよび151Eに示されている。ここで、位置が選ばれ、次にこれらの位置における指紋が算定され、その結果に関してハッシュが算定される。インスタンスに関連したタグ120内に、ソフトウェアインスタンス111の指紋を含めることに

よって、ユーザー装置104～107における管理プログラム（ソフトウェアのアクセスに用いられる、図4の符号209）が、INST_SWとタグの関連付けが正しいことを確認することができる。この関連付けが正しいかは、INST_SWにおいて同一位置指紋検査（詳細な定義は図6による）を行って、関連したタグの指紋リストと比較することによって確認する。指紋の使用がHASH_SWの関数とオーバーラップしてもよく、これによりタグとソフトウェアインスタンスの関連付けが正しいことの確認には、大きな効果が生じる。

【0193】

例えば、百科事典またはビデオのような大きなソフトウェアインスタンスINST_SWに対して、管理プログラムによるINST_SWの全ての走査を必要とするHASH_SWの算定には、相当の時間を要する。INST_SWに関連したタグがタグサーバーにより算定される前記固定位置の指紋値を含むと、管理プログラム（図4の符号209）はINST_SWのこれらの位置にのみアクセスして、該当の指紋値を算定するだけでよい。前記の指紋を用いることによって、追加的な保護上の利点が生じる。これは、タグサーバーによって指紋が算定される位置が、侵害者の権利侵害に応じて、経時点に変化できるからである。

【0194】

ソフトウェアSW全体ではなく、SWの特定の部分のみについてタグサーバー102によってハッシュ関数値HASH_SWが算定される（図3AおよびBのステップ152）ならば、同様の効率および安全性が得られる。タグサーバー102によって指紋が算定されたソフトウェアインスタンスINST_SW111～114における特定の位置は、タグ120内に指紋を明示的に伴うか、またはインスタンスINST_SWもしくは装置104～107の管理プログラム（図4の符号209）に含めてもよい。タグ120にこれら指紋位置を用いることの利点は、送られるインスタンスINST_SWごとに、指紋が変化でき、指紋は固有のNUM_INST_SWの一種として用いられ、ソフトウェアコードが変更していることのランダムチェックを可能にすることである。

【0195】

したがって、以下の領域組み合わせからなるタグ120は、全て本発明の範囲

内に含まれる。；図3A、3Bおよび3Cの処理結果として生成されるタグ。；前記の組み合わせのいずれかに、ID(SP)のようなユーザー装置（例えば104）に対する管理プログラム識別子209-A（図4）を加えたもの。ただし、ID(SP)の値は、ハッシュ関数値HASH_INST_SWの算定に組み合わせられる。；前記の組み合わせのいずれかに、ソフトウェアSWの内容に関連した指紋リストを加えたもの。ただし、これらの指紋の値は、ハッシュ関数値HASH_INST_SWの算定に組み合わせられる。；前記の領域の組み合わせのいずれかの上位セット。前記のタグおよび処理の記述は、本発明の実施形態の特定の例について記載しているが、当業者には、複数の特定のソフトウェアインスタンスの一つの使用を一意に識別および監督するために、本発明によって提供されていると理解される。

【0196】

タグ120がソフトウェアインスタンス111～114に対して作成されると、タグ120は、ステップ156において、タグサーバー102によって保護センターのデータベース（図9の符号129、138に関して後述される）、ユーザー装置104、ソフトウェアベンダー、またはこれら団体の組み合わせに対して安全に送られる。

【0197】

図2に戻って、タグ120は、タグサーバー102によって、一つ以上のソフトウェアベンダー101、保護センター103およびユーザー装置104に安全に配布される。タグ120が、タグサーバー102によってソフトウェアベンダー101には安全に送られるが、ユーザー装置104～107には送られない場合、ソフトウェアベンダー101によってソフトウェアインスタンス111～114と共に、タグ120はユーザー装置104～107に送られる。別の実施形態では、ソフトウェアインスタンス111～114は、タグ120とは別個にユーザー装置104～107によって取得される。タグ120は、タグサーバー102からユーザー装置104～107によって直接取得される。代わりに、タグ120は、一つ以上の保護センターセンタ103から取得することができる。

【0198】

ソフトウェアのインスタンス111～114は、安全に配布されることを要求されないが、それらは本発明のシステム109の代案実施形態には要求されることがある。ソフトウェアのインスタンス111～114の配布は、いくつかの方法で行うことができる。インスタンス111～114は、コミュニケーションネットワーク100を通じサポートされるダウンロード機構を介してソフトウェアベンダーからダウンロードすることができる(図1)。ダウンロード機構の例は、File Transfer Protocol (FTP)、情報を受領者に送るPUSHプロトコル、TCP/IPおよびWorld Wide Web関連のプロトコル、ならびにコンピュータプロセッサの間のバスを介してデータを送るために用いられる他のプロトコル、または例えばインターネットを代表とするコミュニケーションネットワーク100のような他のタイプのコンピュータネットワークである。

【0199】

あるいは、ユーザー装置104は、ソフトウェアベンダー104と同じ固体であることもないこともあり得るユーザー装置メーカー(図示されず)によりあらかじめインストールされるソフトウェアのインスタンス111～114をあらかじめ備えることができる。例は、ユーザー装置104の中のファームウェアに組み込まれたソフトウェアのインスタンス111～114である。他の代案として、ユーザー装置104のユーザー(この図には示されていない)は、ソフトウェアのインスタンス111～114をユーザー装置により読み取り可能な媒体、例えば磁気的に符号化されたハードディスク、またはフロッピー(登録商標)ディスク、または光学媒体、例えばCD-ROM、DVDディスク、ビデオ、またはオーディオテープ、ホログラフィック格納装置、または情報媒体としての別の媒体を通じて購入することができる。ソフトウェアのインスタンス111～114を入手するためのユーザー装置104～107に対する前記の代案のそれぞれにおいて、本発明によれば、ソフトウェアのそのインスタンスを用いるために必要とされる付随タグ120がソフトウェアのインスタンスに直接付随するか、または別個に好ましくは安全に前記装置に伝送されることができる。

【0200】

図2に示されたユーザー装置104は、ユーザー装置結合機構200への結合を含む。ユーザー装置ストレージ200は、ソフトウェアの各インスタンス111～114、タグテーブル210および指紋テーブル126を維持することができる。指紋およびタグテーブル126、210の目的および詳細は、後述される。

【0201】

図4は、本発明により構成されたユーザー装置104の好ましい構成を図示する。ユーザー装置104は、ユーザー装置ストレージ200、プロセッサ201、メモリ202、相互接続機構203、入／出力機構204を結合するインターナルバス206を含む。ユーザー213は、ユーザー装置104と相互に作用し合う。ユーザー213は、好ましくは人間であるが、本発明は、この中に説明されている使用管理が、大きな、人間以外の相互作用環境内の電子要素において実行されるシステムに適用することができる。この図において、ユーザー213は、本発明の目的を明確に示すためにソフトウェアのインスタンス111～114と直接に相互作用していることを示されている。実際には、ユーザー213は、プロセッサ201のコントロールの下でソフトウェアのインスタンス111～114との間で直接入出力を行うユーザー入／出力機構204と、実際にインターフェースすることができる。

【0202】

ユーザー入／出力機構204は、キーボード、マウス、マイクロフォン、スピーカ、モニタ、ヘッドアップ、またはバーチャルリアリティディスプレイ、またはユーザー装置104と相互作用するユーザー213、または他の機構（すなわち人間ではない）との間に情報を交信するために用いられる他の入／出力装置の1つ以上であることが考えられる。入／出力機構204は、またユーザー装置104がそれによりソフトウェアのインスタンス111～114を備える手段として使用されることができる。この場合、入／出力機構204は、CD-ROM、またはDVDドライブ、スキャナ、フロッピーディスク、またはユーザーストレージ装置200、またはメモリ202、またはユーザー装置（例えば、104）に含まれ、または付随することのできるバッファ（図4には示されていない）

に情報をロードするのに用いることのできる他の機構を含むことができる。

【0203】

相互接続機構203は、コミュニケーションネットワーク100にインターフェースするために用いられ、モデム、ネットワークインターフェースカード、ワイヤレスレシーバ、またはコミュニケーションに用いられる装置であることが考えられる。

【0204】

ハード、フロッピー、または光学ディスクドライブ、RAIDアレー、ファイルサーバー、または他の読み／書き格納機構であることの考えられるユーザーストレージ装置200は、本発明により用いられる各種の要素およびデータを保持するのに用いられる。特にこの実施形態に図示されたように、ユーザーストレージ装置200は、ソフトウェアのインスタンス111～114、タグテーブル210、指紋テーブル126、管理プログラム209（図4）およびカーネル208を含むオペレーティングシステム207を保持する。この分野で理解されているようなオペレーティングシステム207は、ユーザー装置104のスタートアップ後に通常メモリ202にロードされ、ユーザー装置104の各種の要素の総合的動作をコントロールするためにプロセッサ201と関連して実行する。あるいは、本発明のオペレーティングシステムおよび要素プロセッサは、本発明を具現化するシステムの構成に組み込むことができる。

【0205】

ユーザー装置104の例は、パーソナルコンピューター、またはワークステーションである。プロセッサ201の例は、Intelベースのプロセッサであり、例えばCeleron、Pentium（登録商標）、PentiumII、PentiumIII、または80X86ファミリ、またはSPARCベースのRISCテクノロジーを用いるプロセッサ、またはMIPSプロセッサである。これらのプロセッサの名称は、代表的なマイクロプロセッサメーカーの商標である場合がある。オペレーティングシステム207の例は、Windows（登録商標）をベースとするあらゆるオペレーティングシステムで、例えばWindows NT、Windows 98、Windows 95、Windows CE、またはW

indows 3.1で、すべてMicrosoft Corporation of Redmond, Washington製であり、またオペレーティングシステム207は、例えば（登録商標）ベースのシステムで、例えばSun Microsystems Inc. of Mountain View, CaliforniaからのSolarisである。ユーザー装置104の他の実施形態は、カスタム、または組み込まれたオペレーティングシステム207をもつ特殊プロセッサ201を用いる専用機であることが可能である。この分野の熟練者は、前述のユーザー装置104がマイクロプロセッサにより制御されるあらゆるタイプの装置であり得ることを理解すべきである。本発明は、図4に示されたユーザー装置104の構成により限定されない。むしろユーザーに対してソフトウェアにアクセスすることのできるあらゆる装置は、本発明の範囲に含まれることを意味している。

【0206】

本発明のシステムの使用管理の構成を提供するために管理プログラム（SP）209が設けられ、オペレーティングシステム207、タグテーブル210、ソフトウェアのインスタンス111～114および選択的に指紋テーブル126（図4）に関連して実行する。管理プログラム（SP）209は、好ましくはオペレーティングシステム207とは別個のものであるが、その延長部分であることが可能である。管理プログラム（SP）209は、また好ましくはいずれかのプログラミング言語（例えば、C、C++、Java、Assembler、または他の言語）で記載され、好ましくはオペレーティングシステム207のある機能にインターフェースし、制御するためにオペレーティングシステム207により設けられるアプリケーションプログラミングインターフェース（API）を使用する。あるいは、組み込まれたシステムでは、ユーザー装置104の中でユーザー装置104、オペレーティングシステム207、管理プログラム（SP）209および他のデータおよび／または要素が電子回路を介して組み込まれ、完全に代行され、またはメモリに格納されることができる。

【0207】

本発明の好ましい実施形態においてユーザー装置104のスタートアップ（す

なわちパワーアップ) するごとにオペレーティングシステム207、管理プログラム(SP)209およびタグテーブル210は、ユーザーストレージ装置200からメモリ202に読み込まれる。ユーザー装置104の最初のスタートアップ時に、好ましくは装置の管理プログラム209(図4)に対する識別番号ID(SP)209-Aは算定され、安全な位置に格納される。前記の用語集において考察されているこの識別番号209-A(テーブル1、ID(SP))は、下記のある組み合わせにより算定される: 利用可能な場合のハードウェア識別番号; 利用可能な場合の保護センター103(図2)により提供される番号; および装置104の中の高精度タイマー(例えばマイクロセカンド)。本発明のシステムにおいて管理プログラム(SP)209は、ソフトウェアのインスタンス111~114とオペレーティングシステム207との間の使用管理インターフェースとして用いられる。管理プログラム(SP)209により提供される使用管理の動作構成が詳細に説明される前に、ソフトウェアのインスタンス111~114および付随のタグ120のユーザー装置104へのインストールが考察される。

【0208】

図5は、本発明の好ましい実施形態によるユーザー装置104に対するソフトウェアインスタンスINST_SWおよび付随のタグTAG_INST_SWをインストールするステップを図示する。タグ120およびソフトウェアのインスタンス111~114の両者は、ユーザー入/出力機構204によりユーザー装置104にロードされることによりインストールされることが可能であり、あるいは相互接続機構203によりコミュニケーションネットワーク100からの受け取りを介して電子的にインストールされることができる。図5におけるステップは、好ましくは本発明の一部として提供される管理プログラム(SP)209コードを実行するプロセッサ201により行われる。管理プログラム209は、例えばカーネル208への延長としてオペレーションシステム207に存在することができるか、またはカーネル208およびオペレーティングシステム207の上の別個の処理として存在し、実行することができる。

【0209】

いずれの場合にもユーザー装置104（この例においてはパーソナルコンピュータであるが、本発明の手段は、発明の主旨に基づくあらゆる他の装置であることができる）は、図5のステップ250における特定の名称をもつソフトウェア（NAME__SW, SW）のインスタンスINST__SWを入手する。ステップ251においては、ユーザー装置104はステップ250において入手される名称のソフトウェアに付随するTAG__INST__SWを安全に入手する。ステップ252においては、本発明のシステムがタグTAG__INST__SWが署名された、または署名されていないタグであるかを確認する。ステップ252は、SIGN__TS関数値がタグTAG__INST__SWの中にあるかないかを確認するために受け取ったタグ情報を調べることにより行うことができる。次に管理プログラムは、下記のようにタグおよびソフトウェアのインスタンスへのその正しい付随を確認することを行う。

【0210】

本発明の好ましい実施形態においては、図3A、3B、または3Cのステップによりタグサーバー102により作り出され、署名されていないタグに対してステップ154A（図3A）により、また署名されていないタグに対して154B（図3および3C）により作り出された内容をもつ。タグTAG__INST__SWが署名されたタグであればステップ（図5の253）は、管理プログラム（SP）209の一部を行うことにより、ハッシュ関数値 $V = \text{HASH}(\text{INST_SW})$ およびハッシュ関数値 $V = \text{HASH}(\text{NAME_SW}, \text{NUM_INST_SW}, V)$ を算定する。管理プログラム209は、次に値UをタグTAG__INST__SWに見られる値 HASH_INST_SW と比較する。2つの比較された値が合致しない時にはタグは無効である。値UおよびVが合致すると管理プログラム209は、タグサーバー102の公開鍵PUBLIC__KEY__TS（図2の116）の使用により、タグTAG__INST__SWの中にあるSIGN__TS（HASH__INST__SW）のデジタル署名を確認する。SIGN__TS（HASH__INST__SW）におけるタグサーバーの署名が確認されない時には、タグTAG__INST__SWは有効ではない。ステップ250において得られた名称のあるソフトウェア（NAME__SW, SW）のインスタンスがステ

ップ253の中でステップ251において得られた無効タグTAG__INST__SWに付随していることがわかれば、ソフトウェアのインスタンスはステップ254において拒絶される。

【0211】

タグTAG__INST__SWが署名のないタグであれば、ステップ257は、署名されたタグの場合に対し前記において用いられた同じステップにより、タグTAG__INST__SWの中にあるハッシュ関数値HASH__INST__SWに対するハッシュ値を確認するために、管理プログラム(SP)209の一部を実行する。HASH__INST__SW値が正しく判定されない時には、タグTAG__INST__SWにエラーがあり、無効タグTAG__INST__SWに付随するステップ250において得られる名称のあるソフトウェア(NAME__SW、SW)のインスタンスは、ステップ254において拒絶される。

【0212】

ステップ254における拒絶は、ユーザー装置104がステップ250および251において得られたソフトウェアINST__SWのインスタンスおよびその付随のタグTAG__INST__SWの廃棄、または除去、または使用を許さないことを、単に意味することができる。ステップ256もまた実行されることにより、ユーザー装置(例えば104)に報復措置を働かせる。ユーザー装置104に対する報復措置は、装置をその後停止し、または機能停止することを含むことができる。報復措置は、この発明の使用管理機能に関してさらに詳しく考察される。

【0213】

ハッシュ関数値および署名SIGN__TS(HASH__INST__SW)は、署名されたタグに対してステップ253において確認される時、または署名されていないタグに対してハッシュ関数値HASH__INST__SWがステップ257において確認される時には、タグに付随するソフトウェアのインスタンスINST__SW(図2の111~114)をユーザーストレージ装置200に格納し、またタグにステータス“INSTALLED”を取り付けられたタグテーブル210にソフトウェアのインスタンス(例えば111)に対する付随のタグTA

G__INST__SWを格納する（後に詳述されるように、図6に詳細に示されたテーブル210の第1欄に）。

【0214】

タグがその中で管理プログラム識別番号ID（SP）209-Aを含む代案実施形態において、管理プログラム209は、タグ120の中の管理プログラム識別番号209-Aがユーザー装置104に格納された管理プログラム識別番号209-Aと同じであることを確かめる。タグ120がその中でソフトウェア内容SWの特定の位置に基づく指紋リストを含む代案実施形態において、管理プログラム290は、指紋がソフトウェアSWにおける同じ特定の位置において計算された指紋と合致することを確認、前記の合致は前記の規定に記載され、この中で詳しく説明されているように同じ位置の指紋に基づいている。

【0215】

図6は、凡例としてのタグテーブル210の内容を図示している。一般にタグテーブル210は、ユーザー装置104のユーザー213、または装置104自体がソフトウェアのインスタンス111～114の使用を許されているかの決定を行うために管理プログラム（SP）209により要求される情報を含む。後述する処理により管理プログラム209は、ソフトウェアのインスタンス111～114の使用の試みを検出することが可能であり、要求されるインスタンス111～114に付随するタグTAG__INST__SWに対する使用管理特性を求めするためにタグテーブル210に保持される情報をチェックすることができる。

【0216】

定期的に管理プログラム（SP）209は、ユーザー装置104を保護センター103（図2）とインターフェースさせる呼び出し手順を実行する。呼び出し手順中、呼び出しを行っているユーザー装置104にインストールされたソフトウェアの各インスタンス111～114に対するタグテーブル210のタグ情報は、保護センター103（図2）の確認プログラム（図9の315）により確認されることにより、ユーザー装置104の管理プログラム209に、ユーザー213が使用を要求しているソフトウェアのインスタンス111に関して使用管理判定を行うことが指示される。

【0217】

図6は、本発明の好ましい実施形態における装置（すなわち104）のタグテーブル210を示す。インストールされた各ソフトウェアのインスタンス111～114に対して、図5のステップ251を介して得られた各有効なタグTAG__INST__SW120は、タグテーブル210の“TAGS”のラベルをもつ第1欄に格納されている。タグテーブル210のTAGS欄の中のタグは、TAG__INST__SW1、TAG__INST__SW2、TAG__INST__SW3、TAG__INST__SW4およびUNTAGGED__SWのラベルをもつ。詳述されるタグテーブル210の中の他の情報は、各タグに対してUSAGE STATUSリスト（第2欄）、ACTION TIME（第3欄）、RUN COUNT（第4欄）およびUSE TIME（第5欄）を含む。管理プログラム（SP）209は、各タグエントリー（すなわち各タグテーブル列）に対するタグテーブル情報を使用することにより、該当タグTAG__INST__SWに付随するソフトウェアの各インスタンス111～114の使用の要求をどう処理するかを決める。

【0218】

簡単に説明すれば、タグテーブル210のUSAGE__STATUS欄は、一般に管理プログラム209に対し、ソフトウェアのインスタンス111～114が、ユーザー213、または装置104～107に対して使用可能か否かを表示する。ソフトウェアの使用が許される時には、ステータス欄は“CONTINUED”、または“INSTALLED”を示すのに対し、使用が否認される時には、この状態は“GC__DISABLED”の語により示される。“REMOVED”ステータス語が次に続く“INSTALLED”は、ソフトウェアのインスタンス111～114に対するタグTAG__INST__SWnが、以前にはユーザー装置104にインストールされていたがもはやインストールされておらず、したがって使用可能ではないことを示す。ACTION TIME欄は、管理プログラム（SP）209（図2）により行われた最後のステータス判断（例えば、説明されるべき、最後の呼び出しおよびタグ確認手順）のタイムスタンプ（例えば日時）を示す。タグテーブル210のRUN COUNT欄は、タグTA

G__INST__SWn（ただし、nはこの例では1から4の数字である）に付随するソフトウェアのインスタンス111～114がユーザー装置104～107に用いられた関数を示す。最後に、タグテーブル210におけるUSE TIME欄は、TAG__INST__SWnに付随するソフトウェアのインスタンス111～114が装置と保護センターとの間の最後の呼び出し手順以来、または他の実施形態ではインストールされて以来経過した使用全時間を示す。

【0219】

各タグ（第1欄）に付随する各種の領域（すなわち列）は、この中で説明された各種の目的のために、この発明のシステムにより用いられる。タグは、付随の列の内容に基づいて特定のソフトウェアインスタンス111～114が適性に、または有効に使用されることができていることを確かめるために調べなければならないタグテーブル210の列を特定するのに用いられる。選ばれた列の現在のUSAGE__STATUS領域は、ソフトウェアインスタンス（すなわちこの列の111～114の1つ）の使用が許されるか否かを判定する。

【0220】

使用が許される時に説明されるように管理プログラム（SP）209は、使用されているインスタンス111～114に対する使用時間およびランカウントを追跡することができる。この情報は、ユーザー装置104～107の事象履歴を構成するために使用することが可能であり、ソフトウェアのインスタンス111～114のpay-per-use、またはpay-per-viewでの使用の追跡のような他の目的にも用いることができる。事象履歴は、すべての試みられた使用、完了した使用、使用時間および装置でのパワーアップのような他の事象のすべての時間的記録である。2つの装置により、それらがたとえ同じソフトウェアインスタンスおよび同じ識別子をもっているとしても同じ事象履歴をもつことは考えられない。

【0221】

ある実施形態において2つの装置は、同じソフトウェアインスタンスおよび同じタグ、または管理プログラム、または装置識別番号をもつことはない。しかし、ソフトウェアに対する狡猾な権利侵害者は、1つの装置のディスクイメージを

別のものに正確にコピーすることを試みることができ、この場合にはタグ、装置および管理プログラム識別子は、正確に複製されることがあり得る。本発明は、ある実施形態においてその識別子（例えばタグ120（図6の第1欄）、装置間のSP ID209-A）に特定のプロセッサ、またはハードウェアシャシを付随させる情報、例えばハードウェアプロセッサ識別番号（すなわち、例えばプロセッサの通し番号）を含めることを固有の識別子の少なくとも1つ（すなわちソフトウェアタグ120、または管理プログラム識別番号209-Aの1つ）にとって可能にすることにより、このような侵害行為の回避を意図する。すなわち、侵害者が全ディスク情報を複製し、複製されたディスクを別の装置に移すことにより、本発明の使用管理保護をくぐり抜けることを試みる時、本発明はハードウェア装置識別機構を情報の中に組み入れ、タグの確認中（すなわち、説明されるべき呼び出し処理中）ハードウェア識別情報がそれにしたがってチェックされることができる。

【0222】

この実施形態は、同じタグ情報を用いることに努める2つの装置を追跡するために、保護センター103（図2）において保持される装置の使用統計値を用いる発明機構を補填すると理解されるべきである。すなわち、侵害者が合法的な装置104からのディスクを他の装置（すなわち107）にコピーする時には、この発明の構成によれば、装置の海賊版107の違法なユーザーにとって、この装置107を、合法的な装置104の使用を正確に複製する方法で使用することはほとんど不可能である。したがって、各装置104、107がタグの確認を行うために保護センター103（図2）に対して呼び出しを行う時に、保護センター103（図2）は、装置104、107の1つを、他の装置（すなわち104、107の他のもの）に関して不整合の使用または呼び出し統計値をもつとして検出する。したがって、各装置104、107が呼び出しを行った場合、装置104、107の1つは、不正のソフトウェアの使用を試みるように見える。この点で本発明のシステムは、継続メッセージ（後述される）に含まれる報復措置を実行することにより、一方または両方の装置、装置のソフトウェア、装置の使用、またはそれらの組み合わせの機能を停止することができる。違法または不法な使

用の正しい機関（例えば法務機関、ソフトウェアベンダー）への報告もまた本発明により行うことができる。

【0223】

pay-per-use、またはpay-per-viewの例としてpay-per-useソフトウェアのインスタンス111～114が用いられるたびに管理プログラム（SP）209は、そのインスタンス111～114に付随するタグTAG__INST__SWに対し、タグテーブル210におけるRUN__COUNT領域（第4欄）にこれを記録することができる。RUN COUNT情報は、後に請求の目的に用いることができる。

【0224】

またタグテーブル210に含まれているのは、この特定のユーザー装置104に対する特定のタグテーブル210を一意に識別するヘッダ領域HEADER__TAG__TABLEである。ヘッダHEADER__TAG__TABLEは、per user 213、またはper user device 104ベースにおいて固有であることが可能である。タグテーブル210がper user 213ベースで固有である時には、ユーザー装置104での各ユーザーアカウント（すなわちloginアカウント）は、そのユーザー213に対して独自のタグテーブル210をもつことができる。per userタグテーブル210は、そのユーザー213のみにより購入されているはずのソフトウェアのインスタンス111～114に対して、タグTAG__INST__SWを使用されるために保持することができる。言い換えれば、唯1つのタグテーブル210が図示されているが、本発明はタグ使用および使用管理を多くのユーザー213において追跡することが可能であり、また各ユーザーは別個のタグテーブル210をもつことができる。

【0225】

HEADER__TAG__TABLEは、好ましくはこのタグテーブル210に対し固有の識別を示すID__TAG__TABLE領域を含む。ID__TAG__TABLE領域は、好ましくは管理プログラムの209ID（SP）209-Aの識別を含む。さらにそれは、このタグテーブル210が付随するユーザー213

ID (USER) の識別ならびにユーザー装置104 ID (DEVICE) (例えば前述のように続き番号、またはhost-id) の識別およびオペレーティングシステム207 ID (OS) の識別を含むことができる。

【0226】

ユーザー識別ID (USER) の例は、ユーザーネームおよび／またはパスワードの組み合わせであることができる。ユーザー装置ID (DEVICE) の識別の例は、ホストネーム、host-id、IPアドレス、続き番号、またはこのユーザー装置を、他のユーザー装置 (例えば図1の104～107) と一意に識別することのできる他のハードウェアまたは装置についての特定の情報を含むことができる。

【0227】

ID (SP) 209-Aは、例えば装置104～107が高精度クロック (図4の205) に基づいて最初に通電される時点に関する情報からなることができる。異なった装置 (すなわち104, 105) からの2つのID (SP)'s 209-Aは、高精度クロック205がマイクロ秒の精度である時には等しくなるのはまれである。ID (SP)'s が等しくなることのリスクを低下させるには、ID (SP) 209-Aが、利用可能ならばハードウェア続き番号および利用可能ならば保護センター103 (図2) からの番号を含むことも可能である。侵害者予備軍にとってディスクイメージをコピーすることは可能であるが、この場合には、2つの装置は同じID (SP) をもつかもしいない。上に簡単に触れたように、またさらに考察されるように、これは、呼び出し中保護センター103 (図2) により捕捉されることができる。オペレーティングシステム207は、ID_TAG_TABLEに領域での識別に用いることができる続き番号、または同等の固有の識別情報をもつこともできる。

【0228】

ヘッダ領域HEADER_TAG_TABLE (図6のタグテーブル210の上列) は、また“最後の保護センター継続メッセージ”領域LAST_GC_CM、“最後の呼び出し時刻”領域LAST_CALLUP_TIMEおよび“装置パワーアップの回数”領域NUMBER_DEVICE_POWERUPSを

含む。さらにヘッダは、事象履歴を扱う二つの領域を含む：現在の事象履歴：HASH (EVENT_HISTORY) および前回の呼び出し時点の事象履歴のハッシュ

HASH (EVENT_HISTORY_AS_OF_MOST_RECENT_CALLUP)。

【0229】

ヘッダのLAST_GC_CM領域 (テーブル210の第1列) は、タグテーブル210更新情報の符号化を含む保護センター (GC) 103 (図2) からの偽造不能のメッセージである継続メッセージ値およびユーザー装置の管理プログラムSPに対するGC103 (図2) により定められた措置および報復措置を含む。タグテーブル210ヘッダにおけるLAST_CALLUP_TIMEは、CALLUP_POLICYにしたがっていつ次のGC103 (図2) への呼び出しが要求されるかを求めるために、管理プログラム209により、他のタグテーブルデータと組み合わせて使用される。NUM_DEVICE_POWERUPSは、いつ呼び出しが必要であるかを定めるための方法の一部として使用される。

【0230】

事象履歴は、装置104～107の各ソフトウェアインスタンス111～114が実行される時、また場合によってはユーザー装置104～107に対する外部入力 (すなわちユーザー装置213相互作用) が起きる時のような情報を含むことができる。事象履歴の目的は、その過去の挙動、または装置の使用に基づき装置104～107を識別することにある。これは、有用であると考えられる。なぜならば、管理プログラム識別番号209-Aおよびタグ120のような静的情報は、1つの装置104～107から他の装置にコピーできるが、事象履歴に具現化された動的情報は、同じ静的情報をもつ装置104～107に対しても発散することが考えられるからである。事象履歴は大きいことがあるから、事象履歴のハッシュ関数値が、事象履歴自体に代わって保持される。好ましくは、2つの事象履歴ハッシュ関数値は、呼び出し手順中に処理を続けさせるために保有される。

【0231】

次に説明されるように好ましくは継続メッセージCM (図2の212: 図13Bの423) は、タグテーブルヘッダのLAST_GC_CM領域にも格納される (図6のテーブル210の上列)。CM212は、ユーザー装置104による呼び出し手順中保護センター103 (図2) により作られるメッセージであり、好ましくは、保護センター103 (図2) により呼び出しを実行する装置104~107に安全に伝送される。継続メッセージCM212は、ユーザー装置104に関する管理プログラム (SP) 209がソフトウェアのいずれのインスタンス111~114が引き続き使用することを許されるか、または不正使用のために機能停止されるべきかを定めることが可能な情報を含み、また装置の管理プログラム209により実行すべき他の措置または報復措置を決定することもできる。

【0232】

LAST_CALLUP_TIME領域は、発生した最後の呼び出し処理 (説明あり) のタイムスタンプを含み、NUM_DEVICE_POWERUPS領域は、ユーザー装置104がパワーアップされた時点の数を含む。次に説明されるように、各ユーザー装置104における管理プログラム (SP) 209は、NUM_DEVICE_POWERUPS、LAST_CALLUP_TIMEおよびLAST_GC_CM継続メッセージのようなヘッダ情報を含むタグテーブル210の正確な情報を維持する (必ずしも作り出すことはない) ことに対して責任を負う。すなわち、継続メッセージ (CM) 212 (図2) は、保護センター103 (図2) により作り出され、ユーザー装置104への管理プログラム (SP) 209に安全に渡される。これを受け取ると管理プログラム (SP) 209は、好ましくは継続メッセージ (CM) 212 (図2) をパースし、タグテーブル210を最新の使用管理情報により更新する (すなわちタグテーブル領域を更新) ことに責任を負う。

【0233】

ヘッダ領域HEADER_TAG_TABLEの中の情報は、一意的にタグテーブル210を識別することが可能であり、ユーザー装置104にインストール

されたソフトウェアの各インスタンス111～114に対する使用管理情報を更新するために使用することができる。この考え方では、各ユーザーまたは各ユーザーおよび／またはユーザー装置104の組み合わせに対するタグテーブル210は、他のユーザー213、または他のユーザー装置104、またはユーザー／ユーザー装置の組み合わせに対する他のタグテーブル210から、HEADER__TAG__TABLEを介して一意的に識別されることができる。

【0234】

ソフトウェアの新しいインスタンス111～114およびその付随のタグ120が図5のステップを介して入手され、インストールされ、使用される時に、そのタグTAG__INST__SWに対するタグテーブル210エントリー（すなわちタグテーブル210の列）は、そのタグに付随するソフトウェアのインスタンス111～114がそのユーザー装置104に新たに追加され、またはインストールされることを示すためのACTIONコラム値を、INSTALLEDにセットされている。ACTION TIME値は空白にされるか、またはインストール時点を示す。RUN COUNTおよびUSE TIMEコラム値はゼロ、または“0”にセットされ、または空白にされる。

【0235】

本発明の別の構成によれば、タグテーブル210に挿入するために作り出されたタグTAG__INST__SW (Column1) を付随させていないソフトウェアインスタンス111～114に対して、使用管理を施すことができる。このようなインスタンス111～114は、ソフトウェアのタグをもたないインスタンス、またはタグなしのソフトウェアとも呼ばれる。タグなしのソフトウェアの例は、ユーザー213により作られたソフトウェアである。ユーザーにより作られたソフトウェアは、ソフトウェアプログラムもしくは歌を書きまたは創出するユーザー213の場合のように、合法的に作り出されることができる。ユーザーにより創出されるソフトウェアは、非合法的に作り出されることもあり、この場合には侵害ソフトウェアINF__SWと呼ばれる。ユーザー装置104～107が合法的なタグなしのソフトウェアを使用することは望ましいことであり、本発明の使用管理はこのような使用を可能にする。しかし、同時に本発明の機構によ

れば本発明は、ユーザー装置104～107がタグありまたはタグなしの侵害ソフトウェアの使用を試みる時には、使用を検出し、阻止し、また必要ならばその装置に対して報復措置を定めることができる。

【0236】

侵害ソフトウェアINF_SWは、例えば次のように作り出されることができるであろう。侵害者ベンダーは、本またはCD-ROMのアプリケーションプログラムのような合法的特定のソフトウェアインスタンス111～114を用い、必要とされるタグ120の関連事項のすべてを、そのソフトウェアに対して含められたインストールプログラムから除去することにより、ソフトウェアインスタンスの海賊版を作り出すことができる。侵害者ベンダーは、次に変更されたソフトウェアのコピー（すなわち付随タグの引用をもはや必要としない）を別の名でタグなしのソフトウェアとして販売するのである。タグなしのソフトウェアの別の例は、ベンダーの本の別の言語への無許可の翻訳、またはアプリケーションプログラムの再コンパイルされたバージョンのような合法的なベンダーのソフトウェアSWの変造、または派生したバージョンとして侵害により作られた侵害ソフトウェアである。本発明のシステムは、ユーザー装置104～107におけるこのような不許可のソフトウェアに対し、使用を防止し、追跡し、防護する。

【0237】

前記を行うために本発明は、指紋と呼ばれるコンセプトを導入する。本質的に指紋は、そのインスタンスに対するソフトウェア（SW）の内容に対して固有の、ソフトウェアのインスタンスに付随する値を作り出す。ソフトウェアのインスタンスの違法に作り出されたコピーの指紋を入手することができる時には、本発明は同様に違法に作られたコピーを使用することの他のユーザー装置104～107による別の試みを検出するための方法を提供する。本発明によればソフトウェアの特定のピースに付随する指紋は、好ましくはユーザー213がユーザー装置104にタグなしのソフトウェアをインストールし、または使用することを試みる時である。

【0238】

図7は、ユーザー装置のタグなしのソフトウェアをインストールする処理を図

示する（この例ではユーザー装置104は考察の中で用いられる）。ステップ330においてユーザー装置213は、タグなしのソフトウェアのインスタンス（すなわち111～114のタグなしインスタンス）をユーザー装置104にインストールする（または作り出す）。タグなしのソフトウェアUNTAGGED_SWは、例えば単にバイナリデータのストリングのように見え（STRING〔0・・・N〕）、最初は付随のタグをもたない。ステップ331のタグなしのインスタンス111～114の使用が試みられると管理プログラム（SP）209は、このソフトウェアのインスタンスに対してタグテーブル210にはタグTAG_INST_SWが存在しないことを検出し、したがって管理プログラム（SP）209は、指紋関数FPを用いてタグなしのソフトウェアインスタンス111～114を指紋付けする。指紋関数は、例えばハッシュ関数であることが考えられる。

【0239】

ステップ331において各指紋 X_i は、好ましくはタグなしのソフトウェアの部分STRING〔i、i+k-1〕において動作する指紋関数FPにより作り出される値に等しい、ただし $0 \leq i \leq m-K+1$ （定数Kに対して）。m個のインデックスを選ぶことが可能である。言い換えれば指紋関数FPは、タグなしのソフトウェアデータSTRING〔0・・・N〕の選ばれたセグメント上で行われる、ただしNはビット単位のタグなしのソフトウェアの全長である。好ましくは指紋関数FPは、複数個の指紋（m）を作り、そのそれぞれは他との間にオフセットを示す。ステップ332において管理プログラム（SP）209は、ユーザー装置104の指紋テーブル210が指紋 X_{i1} から X_{im} を格納する。

【0240】

前記に代わる実施形態においては指紋は、タグなしのソフトウェアの非連続部分に基づいて作られる。

【0241】

別の代わりの実施形態においては指紋は、ソフトウェアが用いられる時にソフトウェアの挙動に基づいて算定される。挙動の例は、ソフトウェアが行うシステムコールのシーケンスであることが可能である。例えば、ゲームソフトは、スク

リンへの書き込みに特別のパターンをもつことができる。これらのパターンは、ソフトウェアのインスタンスの指紋に使用される。

【0242】

最後にステップ327において管理プログラム(S P)209は、ユーザー装置上のタグなしのソフトウェアのインスタンス111~114の存在を示すためのタグテーブル210のタグなしのタグエントリーUNTAGGED_SWを作り出す。タグテーブル210におけるUNTAGGED_SWタグは、指紋付けが行われたソフトウェアのタグなしのインスタンスにタグUNTAGGED_SWを一意的に付随させるために、ハッシュ関数または他の手段を用いることができる。前記の処理を用いることによりユーザー装置104にソフトウェアのタグなしのインスタンス111~114を使用し、またはインストールする試みは、そのタグなしのインスタンスが指紋付けされ、またUNTAGGED_SWタグがタグテーブル210の中に作り出される結果をもたらす。

【0243】

下記に説明されるように指紋テーブル126は、保護センター103(図2)が認めた侵害ソフトウェアINF_SWの使用を検出するために保護センター103(図2)により使用される。この発明の指紋構成の使用の詳細は後述される。

【0244】

図8は、ユーザー213がソフトウェアのインスタンス(INST_SW)111~114をユーザー装置104において使用することを試みる時にこの発明のシステム109により行われるハイレベルステップを示す。ステップ270においてはユーザー213は、ソフトウェアのインスタンス111~114を使用するためにユーザー装置104のユーザー入/出力機構204にインターフェースする。ステップ271においては管理プログラム(S P)209は、ソフトウェアのインスタンス111~114を使用させるためのコールを阻止する。この点で管理プログラム(S P)209は、要求されたタグテーブル210における“CONTINUED”ステータスを示すタグTAG_INST_SWをもつことを確かめる。しかし、好ましい実施形態において個別のタグTAG_INST

__SWnをチェックする前に管理プログラムは、タグテーブル210自身が有効な、または更新された状態にあることを確かめる。有効な状態により意味されることは、タグテーブル210は、期限切れになっておらず、その内容を更新するために呼び出し手順を必要としていることである。したがってステップ272においては管理プログラム(SP)209は、保護センター103(図2)への呼び出しが現時点で必要であるか否かを確かめるために、タグテーブル210をアクセスする。

【0245】

前記に代わる実施形態において指紋がタグにおいて含まれている時には、管理プログラムSP209は、使用されているソフトウェアが同じ位置の指紋を用いることにより、このタグに正しく付随していることをチェックすることができる。

【0246】

呼び出し手順は、有効性を効果的に再確認し、タグテーブル210内での各タグTAG__INST__SWnの使用管理方針を実行するために、本発明のシステムにより定期的に行われる。呼び出し処理は、保護センター103(図2)とユーザー装置104との間で行われる。保護センター103(図2)に対して行われるべき呼び出しを行わせる引き金となる多くの事象の存在が考えられる。

【0247】

例えば、管理プログラム(SP)209によりステップ272において行われる呼び出しの決定は、タグテーブルヘッダHEADER__TAG__TABLEのLAST__CALL__UP__TIME領域を調べることにより行うことができる。LAST__CALL__UP__TIMEにおけるタイムスタンプがある経過時間を越えている時には、保護センター103(図2)に対する呼び出しが必要であり、呼び出しの処理が行われるステップ273に移行することにより行われる。前記の代わりに要求される呼び出しのために満足されねばならない1群の原則、または条件を限定するタグテーブル210自身に対して呼び出し方針(CALL__UP__POLICY)が考えられる。

【0248】

他の実施形態においては、ソフトウェアの個々のインスタンス111~114に付随する呼び出し方針(CALL_UP_POLICY_SW)の存在が考えられる。この場合にステップ272は、ソフトウェアコンテンツSW、またはステップ270におけるユーザー213により要求されたアクセスであったソフトウェアのインスタンス(INST_SW)111~114に付随する呼び出し方針(CALL_UP_POLICY_SW)のルール、またはテストを調べることができる。別の実施形態においてはユーザー装置104のユーザー213がソフトウェアのタグなしのインスタンスを使用することを試みる時には、ステップ272は呼び出しを必須条件とすることができる。別の実施形態においてはユーザー装置104のユーザー213が、初めはタグ付きソフトウェアを使用し、次にステップ272は、呼び出しを必須条件とすることができる。別の実施形態において継続的な呼び出し手順の間に許される最大間隔は、好ましくはユーザー装置104における経過時間、ソフトウェアのインスタンス111~114の使用の回数と使用時間、装置104のパワーアップの回数および／または装置104の時間または使用に関する他の手段の組み合わせにより決まる。

【0249】

呼び出しの処理は後に詳細に考察される。しかし、本質的に呼び出しの処理中は、ユーザー装置104の中の管理プログラム(SP)209が、タグテーブル210および指紋テーブル126のコピーを保護センター103(図2)に安全に引き渡す。確認後保護センター103(図2)は、タグテーブル210の各タグTAG_INST_SW_nを損なわれたタグのリストと比較する。保護センター103(図2)は、ある方法で無効、または損なわれたタグを検出することができる。

【0250】

各タグに付随する使用管理方針POLICY(TAG_INST_SW)は、タグ120(およびしたがってタグに付随するソフトウェアのインスタンス)が使用管理方針POLICY(TAG_INST_SW)に基づいて使用されていることを確かめるために保護センター103(図2)においてチェックすることもできる。方針は、全ユーザー装置104~107、またはper user 2

13、またはper tag 120ベースであることができる。またタグなしのソフトウェアに対しては、指紋テーブル126は侵害ソフトウェアINF_SWの使用を検出するために保護センター103（図2）の指紋データ構造体（後に説明あり）に対して比較することができる。タグテーブル210および指紋テーブル126の分析が完了した後、保護センター103（図2）は、継続メッセージ（SM）212（図2）を作りユーザー装置104に送り返す。

【0251】

前記に代わる実施形態においてタグ付きのソフトウェアは、また指紋によってチェックすることができる。この実施形態は、侵害者ベンダーが、知的所有物または合法的なベンダー（すなわち101）の他の権利を侵害する特定のソフトウェアのインスタンスを、タグ付きすなわちタグサーバー102から入手した合法的なタグを添えられたソフトウェアとして配布することを防止する。この実施形態においてはユーザー装置104～107の管理プログラム209は、タグ付きのソフトウェアインスタンス111～114においても指紋処理を実行し、算定された指紋をその指紋テーブル126に格納する。呼び出し手順中にユーザー装置104～107に用いられるタグ付きのソフトウェアインスタンス111～114から得られる指紋もまた侵害ソフトウェアの使用を検出するために保護センター103（図2）に送られる。

【0252】

継続メッセージ（CM）212（図2）は、ユーザー装置（例えば104）のソフトウェアのインスタンス111～114の動作、または装置104自身の動作に影響を及ぼすことのある各種の情報を含む。例えば、保護センター103（図2）がユーザー装置104に対しタグテーブル210における無効のタグTAG_INST_SWnを検出すると、そのユーザー装置1104に戻された継続メッセージ（CM）212は、ユーザー装置104に対し特定の時間にわたり、または無制限に機能を停止させることができる。あるいは前記に代わり継続メッセージ（CM）212は、ユーザー装置104が無効タグ120に付随するソフトウェアの特定のインスタンス（INST_SW）を使用するのを無効にすることができる。

【0253】

ユーザー装置104において実行される措置は、継続メッセージ(CM)212のACTIONS部分において限定され、後に詳述される。継続メッセージ212は、またタグテーブル210における情報を更新するためにユーザー装置104における管理プログラム(SP)209によって使用される。例えば、そのタグテーブル210のACTION TIMEコラムは、最新の継続メッセージ(CM)212のタイムスタンプにより更新されることができるので、各タグTAG__INST__SWnが保護センター103(図2)により最近チェックされた時点を示すことができる。

【0254】

図8の処理の記述を続ける際にステップ273における呼び出し処理が完了した後タグテーブル210は、ステップ277におけるユーザー装置104において更新され(すなわち継続メッセージ212を介して)、処理はステップ272に戻る。ユーザー装置がこの時点で保護センター103(図2)への呼び出しが不必要と認めた時には、処理はステップ274に進み、ステップ273においてユーザー213により使用が要求されたソフトウェアの特定のインスタンス111~114の使用状況が求められる。

【0255】

ステップ274においてユーザー装置104の管理プログラム(SP)209は、要求されたソフトウェアのインスタンス111~114が付随するタグTAG__INST__SWnに対するタグテーブル210におけるUSAGE STATUSコラムを重点的に調べる。USAGE STATUSコラムが“CONTINUED”を示す時には、管理プログラム(SP)209は、オペレーティングシステム207のカーネル208に信号を送ることにより、ステップ275の要求されたソフトウェアのインスタンス111~114の使用を許す。要求されたソフトウェアのインスタンス111~114に付随するタグ(TAG__INST__SWn)に対するタグテーブル210におけるUSAGE STATUSコラムが“GC_DISABLED”、または“REMOVED”を示す時には管理プログラム209は、ステップ276でのソフトウェアのインスタンス111

～114の使用を否認する。

【0256】

要求されたソフトウェアのインスタンス111～114に使用が許されると管理プログラム（SP）209は、要求されたソフトウェアのインスタンス111～114が付随するタグTAG__INST__SW_nに対するRUN COUNTコラムにおける値を1つだけ増やす。管理プログラム（SP）209は、要求されたソフトウェアのインスタンス111～114が使用されている時間を追跡し、これに基づいてタグに対してUSE TIMEコラムを更新する。

【0257】

図9は、保護センター103（図2）の構造の好ましい実施形態を図示する。保護センター103（図2）は、プロセッサ301、メモリー302、相互接続機構303、クロック304および保護センター承認データベース300を結合するバスを含む。保護センター103（図2）は、好ましくは1回に複数の処理のための多くの操作を行うことのできるマルチプロセッササーバーのようなハイパワードコンピュータシステムである。相互接続機構303は、例えば保護センター103（図2）がコミュニケーションネットワーク100を介して同時に多くのユーザー装置104と通信することを可能にするモデムバンク、または1つ以上の高帯域幅ネットワーク接続である。

【0258】

保護センター103（図2）の承認データベース（GCDB）300は、好ましくは広汎な量の情報を格納する能力もつ大きいデータベースサブシステム、またはディスク、またはRAIDアレーである。この実施形態においてGCDBは、タグ付きのソフトウェアのインスタンスに対するデータおよび指紋データ構造体137に対するデータを保持するタグ付きのソフトウェアデータベース138（図9）を含む。タグ付きのソフトウェアデータベース（図9）は、各ユーザー装置104のソフトウェアの各タグ付きインスタンスに対する呼び出し記録（図10の320、321）を含む。これらのデータベース137および138（図9）のそれぞれの内容と使用は詳述される。

【0259】

保護センター103（図2）の動作中メモリ302は、プロセッサ301と関連してこの中に記された保護センター機能を実行する確認プログラム（VRP）315を格納するのに用いられる。メモリ302は、またはタグの確認および上に簡単に説明された呼び出し手順中のタグ確認および使用管理決定のために、保護センター103（図2）に移されるユーザー装置タグテーブル210および指紋126に格納する。

【0260】

図10は、タグ付きのソフトウェア（例えば111～114）各インスタンスに対し保護センター103（図2）でタグ付きのソフトウェアデータベース138（図9）に維持されるデータ構造体320、321を示す。タグデータ構造体320は、ソフトウェアの各インスタンス111～114に対するタグ120が作られた時に、タグサーバー102から保護センター103（図2）に最初に送られる。好ましくはタグサーバー102から保護センター103（図2）にタグ120が与えられる方法は、コミュニケーションネットワーク100を通じ電子的で安全な配布によるものである。前記に代わり、ソフトウェアベンダー101は、ユーザー装置104～107に配布されるソフトウェアの各インスタンス111～114に対するタグ情報を常に知られていることを保障することの責任を負うことができる。

【0261】

タグデータ構造体320は、ユーザー装置104に用いられるソフトウェアの各インスタンスに対するタグ付きのソフトウェアデータベース138（図9）にある。図示されるように各タグデータ構造体320は、各種の領域を含む。これらの領域は、ソフトウェアのそのインスタンスに対するタグTAG__INST__SW、そのソフトウェアに対する使用管理方針POLICY（TAG__INST__SW）およびソフトウェアのそのインスタンスに対する1つ以上の呼び出し記録CALL__UP__RECORDn321に対する関連事項のリストを含む。

【0262】

ソフトウェアのインスタンス111～114に対するタグTAG__INST__SWnに付随する方針POLICY（TAG__INST__SW）は、ソフトウェア

アベンダー101、または別の組織により定められ、使用権、またはタグに付随するソフトウェアのインスタンスに対するpay-per-useアクセス制限の防護に関する規制と方針を規定する。例えばソフトウェアの特定のインスタンス111～114に付随するタグデータ構造体320に対しては、POLICY (TAG__INST__SW) データは、ソフトウェアのインスタンスを使用するごとにユーザー装置104が定められた料金を支払わねばならないことを内容とする規制を含むことができる。

【0263】

呼び出し処理（後述される）の間、保護センター103（図2）がユーザー装置104からタグテーブル210を受け取る時に、そのユーザー装置104によりソフトウェアの特定のインスタンス111～114が使用された回数は、タグテーブル210のソフトウェアのそのインスタンスに対するタグTAG__INST__SWに付随するタグTAG__INST__SW_nのRUN COUNTコラムから求めることができる。保護センター103（図2）は、次にタグ付きソフトウェアデータベース138（図9）のそのタグTAG__INST__SW_nに付随するタグデータ構造体320に対する方針POLICY (TAG__INST__SW) に依存することができる。保護センター103（図2）は、タグテーブル210の中のRUN COUNT領域により示されている使用回数が前の呼び出し処理から得られた前の数よりも多いか否かを調べることをできる。数が多ければ、保護センター103（図2）は、ユーザー装置104のオーナーまたはユーザー213に送られるべき請求用のこの情報を記録することができる。

【0264】

他の使用管理方針POLICY (TAG__INST__SW) は、保護センター103（図2）がソフトウェアの特定のインスタンス111～114に対する特定の数の使用のみを可能にするように規定されることができる。使用回数を上回ると保護センター103（図2）は、ソフトウェアの前記のインスタンスにタグの付随したユーザー装置のタグテーブル210に付随するUSAGE STATUS領域を値“GC__DISABLED”に送られることができる。タグテーブル210の分析後、そのユーザー装置104に保護センター103（図2）に送

られる継続メッセージ (CM) 212に適切な情報を記載することにより、ユーザー装置104において変更を行うことができる。ユーザー装置104が機能を停止されているタグTAG__INST__SW_n (すなわち図6のタグテーブル210のTAG__INST__SW3) に付随するソフトウェアのインスタンス111~114を使用することを試みる時には、図7において上に説明されているように使用は否認される。

【0265】

保護センター103 (図2) の中のタグ付きのソフトウェアデータベース138 (図9) における各タグデータ構造体320は、図10に示された呼び出し記録CALL__UP__RECORD_n321への引用回数を含む。呼び出し記録CALL__UP__RECORD_n321は、呼び出しタイムCALL__UP__TIME、コールするユーザー装置104のタグテーブル210からのヘッダ領域HEADER TAG__TABLE、タグテーブル210HASH (TAG__TABLE) のハッシュ関数値、およびACTION領域を含む。したがって送られたタグの数に関係なく呼び出しごとに1つのCALL__UP RECORDが生じる。

【0266】

CALL__UP__TIME領域が現在のCALL__UP__RECORD_nに対する呼び出しのタイムスタンプを示す。HEADER__TAG__TABLEは、呼び出し手順_n中のコールするユーザー装置104から受け取られるこのタグデータ構造体320に対するTAG__INST__SW_nを含むタグテーブル210のタグテーブルヘッダを含む。HASH (TAG__TABLE) 領域は、タグテーブル構造体320に付随するタグTAG__INST__SW_nを含んでいたタグテーブル210の中のデータのすべてについて算定された実現不能なハッシュ関数値を含む。最顕にACTION領域は、タグデータ構造体320に対してタグTAG__INST__SWに付随するソフトウェアのインスタンス111~114に対して実行されるべき呼び出し手順_n中に保護センターにより定められた措置をリストする。ソフトウェアの各インスタンス111~114に対するタグデータ構造体320を用いる時に保護センター103 (図2) は、ユーザー装置1

04を介して用いられるソフトウェアのインスタンス111～114に対する使用管理機構に関連する詳細な情報を維持することができる。

【0267】

図11は、保護センター103（図2）の中に維持されている指紋データ構造体137を作る結果となる処理ステップを示す。前記の、図7に関して説明されたように指紋は、タグなしのソフトウェア、および場合によってタグ付きソフトウェアもまた最初にユーザー装置104に用いられる時に、各ユーザー装置104の中の指紋テーブル126に格納される。本発明によればソフトウェア構成は、ベンダーソフトウェアをコピーし、タグの確認を要求するソフトウェアの部分を取り外し、または合法的なソフトウェアの派生体を作り配布することにより、合法的なベンダーの権利を侵害することが考えられる。このように作られたソフトウェアは、侵害されたソフトウェアINF_SWと称される。保護センター103（図2）の中に作り出された指紋データ構造体137は、権利侵害のソフトウェアのインスタンスINF_SWについて算定された指紋を含む。

【0268】

ステップ340の図11においてソフトウェアベンダー101は、権利侵害のソフトウェアのインスタンスの存在を検出する。ステップ341においてソフトウェアベンダー101は、権利侵害のソフトウェアのインスタンスINF_SWのコピーを保護センター103（図2）に提出する。侵害ソフトウェアは、STRING_INF[0・・・N]と見えるバイナリ数字（ビット）のストリングであるに過ぎない。ステップ342においては保護センターは、ユーザー装置104のそれぞれについて管理プログラム（SP）209が指紋を算定するために用いるのと同じ指紋公式FRを用い、権利侵害のソフトウェアのインスタンスにおける指紋Y_iの収集物を算定する。すなわち一連の指紋Y_iは、下記のように計算される。

【0269】

$$Y_i = FR(STRING_INF[i, i+k-1])$$

【0270】

ただし、 $0 \leq i \leq n-k+1$ 、 $n-k$ は、算定する指紋の数である。次に

ステップ343において保護センター103(図2)は、算定された指紋 Y_1, \dots, Y_{n-k+1} のそれぞれをGCDB300の中の指紋データ構造体137に使用する。前記に代わる実施形態において指紋は、STRING_INFのノンコンセクティブシーケンスにおいて算定され、このシーケンスはINF_SWに対して一意的、またはほぼ一意的である。

【0271】

指紋処理は、保護センター103(図2)において完了し、侵害ソフトウェアINF_SWは破棄されるか、または他のコミュニケーションネットワーク100のような他の場所での他の保護センター103において利用することができる。

【0272】

この時点でユーザー装置104に関する管理プログラム(SP)209が、タグのない(およびおそらく権利侵害する)ソフトウェアのインスタンスを使用することの要求を検出すると、管理プログラム(SP)209は、UNTAGGED_SWの指紋を記録する。後にSP209がタグテーブル210および指紋テーブル126を保護センター103(図2)に移すための呼び出し手順を実行する時には、記録されたUNTAGGED_SWの指紋が送られる。ある実施形態においてはタグなしのインスタンスを使用するためのユーザー装置104~107におけるアクセス要求は、呼び出しを発生させることがある。一般位置の指紋を用いることにより指紋テーブル126の指紋は、保護センター103(図2)の指紋データ構造体137における指紋と比較することができる。ソフトウェアインスタンスUNTAGGED_SWが保護センター103(図2)により感知され、それ自体に指紋付けを行った権利侵害のソフトウェアインスタンスINF_SWのコピーである時には、このことは検出され、報復措置がユーザー装置104において継続メッセージ212の返却により行うことができる。別の実施形態においてはユーザー装置104のUNTAGGED_SWのシステムコールの挙動(すなわちシステムコールのシーケンス)は、保護センター103(図2)のINF_SWについて予測されるシステムコール挙動と比較される。他の実施形態において最後の2パラグラフに詳述されたステップは、タグ付きのソフトウ

エアの使用に対するユーザー装置に関する要求の場合にも適用される。

【0273】

この発明の指紋構成とは別に、次に説明される呼び出し手順中保護センター103（図2）の確認プロズラム315は、タグテーブル210の情報を使用管理決定を行うためにタグソフトウェアデータベース138（図9）の情報と比較する。

【0274】

図12は、本発明の好ましい実施形態での呼び出し手順を実行するためにユーザー装置104において実行される管理プログラム（SP）209により行われるステップを図示する。図12のステップは、図8のステップ273の中で行われる。

【0275】

図12におけるステップ370において管理プログラム（SP）209は、保護センター103（図2）を呼び出す。呼び出しとは、ユーザー装置104の管理プログラム（SP）209が、保護センター103（図2）にコミュニケーションネットワーク100を介して接続され、またはメッセージを交換することを意味する。好ましい実施形態においては管理プログラム（SP）209は、HEADER__TAG__TABLEを保護センター103（図2）に送る。保護センター103（図2）は、装置のID__TAG__TABLEからなる前の継続メッセージ、最後の呼び出しの時間LAST__CALLUP__TIMEが、この同じHEADER__TAG__TABLEをもつ最近のCALL__UP記録のCALLUP__TIMEに等しくない限り、呼び出しエラーを生じる。この実施形態の利点は、たとえ数台の装置104～107が同じID__TAG__TABLE（図6のタグテーブルの列1）および同じタグ210（普通侵害のために生じる）をもっている、これらの同じ装置は同じ継続メッセージを受け取っても下記の理由で正しく受け入れることはなく、したがって唯1つの装置（すなわち104～107の1つ）が特定のHEADER__TAG__TABLEを送る。

【0276】

ユーザー装置104～107におけるソフトウェアのインスタンス111～1

14をユーザーが使用することを試みる場合には、上に説明された通り、CALL__UP__POLICY、またはCALL__UP__POLICY (TAG__INST__SW) にしたがって呼び出しが行われる。すなわち、ユーザー装置104のCALL__UP__POLICYによる次の呼び出しの前に時間が許されるか、またはそのインスタンスに対するソフトウェア (SW) のCALL__UP__POLICY (TAG__INST__SW) が経過したソフトウェアのインスタンス111~114の使用をユーザー213が試みる時には、その装置104~107の管理プログラム209がステップ370を開示する。別の実施形態においてSP209は、ソフトウェアのインスタンス111~114の使用が要求されているか否かには関係なく、時間の経過する前の選ばれた時点に呼び出し手順を実行する。CALL__UP__POLICYは、ユーザー装置104の管理プログラム209の中に保持されることができる。さらに使用の要求とは関係なく実行される管理プログラム209の部分が、呼び出しを行う時点であることを求めることにより呼び出しが行われることが可能である。例えばユーザー装置104~107のある回数のBOOTUPS (パワーアップ) が行われた結果、またはタグなしのソフトウェアの最初の使用の結果起きることが考えられる。

【0277】

ステップ371の保護センター103 (図2) に対する呼び出しが失敗すると、処理は、ユーザー装置104の管理プログラム (SP) により報復措置の行われることのあるステップ376に進む。好ましい実施形態において管理プログラム (SP) 209は、新たな呼び出しを行い、報復措置を始める前に数回試みる。報復措置がステップ376において必要である場合には、報復措置は、単にユーザー213に要求されたソフトウェアのインスタンス111~114が、通信のエラーにより一時的にアクセス不能であることを通告するにとどめることが可能である。

【0278】

呼び出しが成功し保護センター103 (図2) にユーザー装置104から接続が行われると、372において管理プログラム (SP) 209は、好ましくはユーザー装置104から保護センター103 (図2) にタグテーブル210を安全

に送るか、または送信する。代案の実施形態においては管理プログラム（SP）209は、指紋テーブル126を保護センター103（図2）にも送る。すなわち、本発明の指紋構成は、ユーザーにより作られ、またはユーザーにより変造された侵害ソフトウェアを検出するために実施形態に使用され、または使用されないことがある。

【0279】

ステップ372が完了した後、管理プログラム（SP）209は、継続メッセージ（CM）212が保護センター103（図2）との間にやりとりの行われるまでは待ちの状態に入る。前記の代わりに管理プログラムSP209は、ステップ372が完了した後にスリープ状態に入り、オペレーティングシステム（OS）207から中断された後に再び走行する。前記に代わる実施形態において管理プログラムSPは、ユーザーからの処理要求に続く。保護センター103（図2）呼び出し処理は、図13Aおよび13Bに関して後述される。保護センター103（図2）が呼び出し手段処理を終えると、継続メッセージ（CM）212がユーザー装置104に送られる。

【0280】

ステップ373においては、管理プログラム（SP）209は、ユーザー装置104の呼び出し方針CALL_UP_POLICYに規定される継続メッセージ212の戻りをチェックする。呼び出し方針CALL_UP_POLICYの中の継続メッセージ（CM）212に対するチェックの例として、ステップ373は、継続メッセージ（CM）212を受け取る前にある経過時間を上回っていないことを確かめることができる。継続メッセージ212の受け取り前に著しく時間が経過している場合には、呼び出し方針は背反されることがある。

【0281】

継続メッセージ212のデジタル署名を確認することのできるような呼び出しの背反が起きるか否かを調べるために他のファクターを用いることができる。呼び出し背反を決める別のファクターは、継続メッセージ212のHASH（EVENT_HISTORY）領域が最後の呼び出しの時点でユーザー装置104に記録された事象履歴のハッシュ、HASH（EVENT_HISTORY_AS_

OF__MOST__RECENT__CALLUP)とは同じでないことである。同じ構成およびID__TAG__TABLEをもつ2つの装置が侵害行為の結果存在するが、1つのみが呼び出しを行う場合には前記は起こり得る。事象履歴のために装置104~107の1つのみが継続メッセージ212を受け取る。他の装置は、自己の呼び出しを行うだけであり、このことが呼び出しのエラーを招く。なぜならば、HEADER__TAG__TABLE(図6のテーブル210の第1列)は、ID__TAG__TABLEで合致するか、前記の理由で呼び出し時点で合致できないからである。

【0282】

CALL__UP__POLICYがステップ373において背反される時には、処理はステップ376に進み、また報復措置がユーザー装置104において行われることができる。この場合には、報復措置は、呼び出しが進むことができず、また要求されたソフトウェアのインスタンス111~114が一時的にアクセスを否認され、または機能を停止されねばならないことをユーザー213に通告することを含むことができる。前記に代わりユーザー装置104は、ある時間機能を停止されることができる。

【0283】

ステップ373が継続メッセージ(CM)212が受け取られ、ステップ374においてCALL__UP__POLICYに規定された限界内にあるために受け入れられることを認めると、継続メッセージ(CM)212が管理プログラム(SP)209に引渡される。次にステップ375において管理プログラム(SP)209が継続メッセージ212を、デジタル鍵署名技術を介して確認し、ユーザー装置104のタグテーブル210における各タグTAG__INST__SW_nに対する継続メッセージ212における各措置を実行する。すなわち、管理プログラム(SP)209は、タグテーブル210の各タグTAG__INST__SW_nに対するUSAGE STATUSおよびACTION TIME欄を更新する。この方法で本発明のシステム109は、ユーザー装置104が定期的にタグテーブル更新210を保護センター103(図2)から得ることが可能になる。

【0284】

管理プログラム（SP）209がユーザー213とユーザー装置104上のインストールされたソフトウェアのインスタンス111～114との間のインターフェースとして用いられるから、管理プログラム209は、好ましくはユーザー装置104の中に記載された使用管理機構を実行する。ソフトウェアのインスタンス111～114に対するタグTAG__INST__SWnが呼び出し処理中にしか変更できない“CONTINUED”使用ステータス状態であることを要求することにより、使用管理は究極的に1つ以上の保護センター103（図2）により管理される。保護センター103（図2）は、ユーザー装置104に対するタグテーブル210におけるタグが、タグおよび指紋のために規定された方針通りに“CONTINUED”または“GC__DISABLED”状態になければならないか否かを決定することの責任を負う。

【0285】

図13Aおよび13Bは、本発明の好ましい実施形態による呼び出し処理中に保護センター103（図2）の確認プログラム（VRP）315により行われるステップを示す、1つの連続フローチャートを示す。保護センター103（図2）は、ユーザー装置104（すなわち、管理プログラム209）が最初の呼び出し処理接続、または図12のステップ307の保護センター103（図2）との接触を行う時には、呼び出し手順を知らされる。これに応じて図13Aのステップ410では、確認保護センター103（図2）がタグテーブル201を受け取る。保護センター103（図2）は、またインストールされるがタグテーブル210の中のタグTAG__INST__SWnにより、タグを施されていないソフトウェアがユーザー装置104にある時には、ユーザー装置104から指紋テーブル126を受け取る。この場合にも本発明の指紋構成は選択的であるが、本発明の好ましい実施形態には設けられている、なぜならば、それらは侵害ソフトウェアの検出を可能にするからである。

【0286】

前記に代わる実施形態においては保護センター103（図2）は、タグテーブル210のみの一部、例えばHEADER__TAG__TABLEおよびタグテーブル210の中のタグ（コラム1）の一部のみを受け取る。受け取られたタグ1

20は、保護センター103（図2）が要求するか、または無作為に選ばれるか、またはその時点でのソフトウェアのインスタンスの使用のためにユーザー装置が必要とする唯一つのタグ120であることが可能である。別の可能性は、タグ120がpay-per-useであるか、または一定の使用数をもつソフトウェアのインスタンスに相当することである。この代案の利点は、それが通信コストおよび処理コストとともに引き下げることである。

【0287】

別の代案実施形態においては、保護センター103（図2）がHEADER_TAG_TABLE（図6のタグテーブル210の上列）のみを受け取る。この実施形態は、保護センター呼び出しを割安にし、各TAG_INST_SWが後述のようにID_TAG_TABLE欄を含む時に十分に機能することができる。次にステップ411の図13Aに関する呼び出し処理の記述に戻る時、保護センター103（図2）は、ユーザー装置104に付随する呼び出し方針CALL_UPPOLICYにしたがっていることを確かめるためにチェックする。ユーザー装置104～107に対する呼び出し方針CALL_UPPOLICY（S）は、好ましくは保護センター103（図2）において維持され、および／またはソフトウェアベンダー101、またはユーザー装置メーカー（図示されず）から必要に応じてユーザー装置104がそのタグテーブル210を確認し、更新しなければならないかを求める方法を、保護センター103（図2）にインストラクトするために提供されることができる。

【0288】

ステップ411は、例えばID_TAG_TABLE領域に含まれるタグテーブル210の固有の識別のようなHEADER_TAG_TABLE情報領域を用いて行うことができる。呼び出しがCALL_UPPOLICYにしたがっていない時には、ステップ416は、継続メッセージ（CM）212が保護センター103（図2）からユーザー装置104に戻される時に管理プログラム（SP）209により実行されるための特別の報復措置を作る。

【0289】

ステップ416および411の両方からステップ412に処理が進み、この点

で確認プログラム315は、タグテーブル210の署名され、および／または署名されていないタグTAG__INST__SW_nを確認する。ステップ412において行われる確認は、タグテーブル210の署名されているタグTAG__INST__SWに対するデジタル署名確認であることができる。署名されていないタグに対してHASH__INST__SW値は、タグTAG__INST__SWの中の秘密番号NUM__INST__SWがそのタグに対するHASH__INST__SWに一致することをチェックするために、用いることができる。これは可能である。なぜならば、HASH__INST__SWは、NUM__INST__SWから部分的に算定されるハッシュ関数値であるからである。さらにNUM__INST__SWは、SPARSE__SETに見出されなければならない、またTAG__INST__SWのNAME__SWに付随しなければならない。

【0290】

ステップ412において検出された、確認されていない各タグTAG__INST__SW_nに対してステップ417は、確認されないタグTAG__INST__SW_nに対するソフトウェアのインスタンス111～114に付随する使用管理方針POLICY (TAG__INST__SW) に基づいて特定の報復措置を作る。この場合の報復措置は、ユーザー装置104の機能を停止するためのインストラクションを含むことができる。ステップ417に定められた報復措置は、それがユーザー装置104に通信された後に実行されることに留意すべきである。

【0291】

ソフトウェアのインスタンス111～114に付随する使用管理方針POLICY (TAG__INST__SW) は、保護センター103 (図2) に保持され、ソフトウェアベンダー101により作られる各種のソフトウェアのインスタンス111～114に対する使用管理の扱い方を保護センター103 (図2) にインストラクトするために、必要に応じてソフトウェアベンダー101から提供することができる。すなわち、ソフトウェアベンダー101は、ソフトウェアのインスタンス111～114を104～107 (例えば料金として) に提供することができる。これらのインスタンス111～114に使用制限を実行するためにソフトウェアベンダー101は、インスタンス111～114に対する方針PO

LICY (TAG__INST__SW) を作り出すことができる。呼び出し手順中保護センターは、方針CALL__POLICY (TAG__INST__SW) を実行するか、または取り締まる。前記に代わる実施形態としてソフトウェア（すなわち111）のインスタンスに対する方針は、111および112が同じソフトウェア内容をもつと仮定すると、その同じソフトウェアのその別のインスタンス（すなわち112）とは相違することが考えられる。これにより本発明は、使用管理を、例えば同じプログラムの2つのユーザーに対して別々に実行することが可能である、なぜならば各インスタンスは独自の付随タグをもち、呼び出し方針はインスタンスごと、またはユーザーごとに別々に維持されることができるからである。

【0292】

いずれの場合にも保護センター103（図2）においてタグテーブル210における各タグTAG__INST__SWが真正であることを確かめられ（ステップ412）、または報復措置が確認されていない各タグに対して作成され（ステップ417）た後に、処理はステップ413に進み、このステップではタグテーブル210における確認された各タグTAG__INST__SW_nが、タグをもつソフトウェアデータベース138（図9）に対比してチェックされる。本質的にステップ413は、ユーザー装置104に用いられたソフトウェアのインスタンス111～114に付随するタグテーブル210における各タグTAG__INST__SW_n（すなわち呼び出し処理を行うユーザー装置）は、ソフトウェアのインスタンスの使用管理方針POLICY (TAG__INST__SW) にしたがって使用されている。各タグがステップ413においてテストされた後、処理はステップ414に進む。

【0293】

ステップ413において実行されるチェック処理は、多様な方法で行うことができる。1つの実施形態によればタグ付きのソフトウェアデータベース138（図9）は、タグTAG__INST__SW_nと管理プログラム識別番号（209-A s）との間の結合のリストおよびこれらの結合が発見された時点を含む。この実施形態において確認プログラム（VRO）315は、タグテーブル210のタ

グと同じタグ120（テーブル210のColumn1）が2つの装置にあるか否かを確認するためのTAG_INST_SW_HEADER_TAG_TABLE_CALLUP_TIME結合のリストに対して比較することができる。タグ120がいくつかのHEADER_TAG_TABLEsに付随することが判明する時には、報復措置をステップ418において作ることができる。

【0294】

本発明の好ましい実施形態においては、保護センターの確認プログラムVRP315は、タグ120 TAG_INST_SWと関連したデータ構造体（図10の320, 321）を採用しており、タグ120と関連したソフトウェアのインスタンス111～114が、呼び出し中のユーザーの装置104において、ソフトウェアのインスタンス111～114に対し指定された使用管理ポリシーPOLICY（TAG_INST_SW）と一致して使用されているかをチェックする。もし、2つの異なるユーザーの装置（例えば、104と105）にはありえない、同一のソフトウェアのインスタンスが、（すなわち同一のタグが）、同時に使用可能な状況（例えば、USAGE STATUS=CONTINUED）にあることを、使用管理ポリシーが指定すると、そのタグのための呼び出し記録321中の詳細なデータが、VRP315によるポリシー違反があったかのチェックを可能にする。

【0295】

タグテーブル210中の各タグ120 TAG_INST_SWが、ステップ413でチェックされると、タグテーブル210中のタグ120は、それらのタグと関連して指定されている報復措置と関係することがある。もし、不正にコピーされたタグ、または、使用管理ポリシーと一致して使われていないタグのために、報復措置が指定されると、処理はステップ420に進み、そこで、保護センター103（図2）中の確認プログラムVRP315が、指定された報復措置を用意し、継続メッセージ（CM）212を介して、ユーザーの装置104に送り返す。このような継続メッセージ（CM）212は、ユーザーの装置104に報復措置を課すために用いられ、ポリシーPOLICY（TAG_INST_SW）違反でのタグテーブル210中のすべてのタグTAG_INST_SW_nのU

SAGE STATUS領域のための“GC_DISABLED”措置値を含んでいる。

【0296】

タグ付けされたソフトウェアのもう1つの扱いでは、上述のタグ処理は、タグテーブルの一部においてのみなされる。例えば、アクセス（すなわち、使われようとしているソフトウェアのインスタンス）を要求しているユーザーの装置104～107（すなわち、ユーザーの装置の管理プログラム209）に対するタグについてのみ、処理がなされる。この場合、継続メッセージ212は、保護センター103（図2）で処理されたタグに関連したソフトウェアのインスタンスについてのみ、継続または報復措置を指定する。

【0297】

さらに他の実施形態では、購入されて無制限に使用されるソフトウェアに対して、タグ処理は全く必要とされず、これにより、ステップ372（図12）に関連した措置がなくなる。代わりに、HEADER_TAG_TABLEのみが、確認される必要がある。この場合、HEADER_TAG_TABLE（図6の上の列）は、ID_TAG_TABLEおよび事象履歴を含む（図6）。この実施形態では、各タグ120は、HASH_SW, NAME_SWおよびNUM_INST_SWに加えて、ID_TAG_TABLEを含んでいる。ID_TAG_TABLE値は、購入時にタグ120（第1欄）に書き込まれ、図3A, 3Bおよび3C中のステップ153のハッシュ関数における独立変数となり、つまり、HASH_INST_SWになる。ID_TAG_TABLEはID(SP)209-Aを含み、ID(SP)209-Aが、装置104が最初に起動されたときに、例えばミリ秒値の時間を含む複製値に基づくことはごくまれであるので、各ID_TAG_TABLE値は、権利侵害のない1つの現実の装置にのみ生じる。

【0298】

ディスクイメージのコピーという形態で権利侵害が起こると、1つのID_TAG_TABLE値がいくつかの現実の装置に生じる（「双子」になる）が、装置104のHEADER_TAG_TABLEのLAST_CALLUP_TI

M領域と、保護センター103（図2）の認証データベース138（図9）のCALLUP_RECORDのCALLUP_TIMEとが、呼び出し時に一致せず、HEADER_TAG_TABLEでの確認が失敗する。これにより、保護センター103は、2つの呼び出しメッセージが2つの別々に構成された装置104～107から送られた場合には、報復措置をとる。

【0299】

さらに、2つの装置104～107は、同一の呼び出し手順を共有しようとすることもできない。なぜなら、それらのHEADER_TAG_TABLEsは、それらのタグテーブル210それぞれのHASH（EVENT_HISTORY）領域によって異なるからである。そのハッシュ関数値が継続メッセージに送られるので、装置104から107のうち1つのみが、継続メッセージ212を適切に処理できる。2つの装置が複製により動作する場合には、管理プログラム209は、このように複製が企てられたことを認識し、報復措置をとる。したがって、各ID_TAG_TABLEは、ただ1つの装置104～107にあり、または関連し、でなければ、呼び出しは失敗する。タグがID_TAG_TABLEを含むとき、装置104～107の管理プログラム209は、タグ120に関連したソフトウェアのインスタンス111～114が、タグ120のID_TAG_TABLE値が適切な装置のそれと一致する場合にのみ、使用されることを許容する。結果として、各ソフトウェアのインスタンス104～107は、ただ1つの装置104～107でのみ使用され、その装置は、タグ120のID_TAG_TABLE値と一致するID_TAG_TABLE値をもつ。

【0300】

ステップ414において、確認プログラム（VRP）315は、タグのないソフトウェアのインスタンスに対してタグテーブル210にエントリーがないか、決定する。ユーザーの装置104～107にインストールされた、タグのないソフトウェアのインスタンスは、特別のタグUNTAGGED_SWで示され、タグのないソフトウェアのためのUSAGE STATUS欄は、UNTAGGEDにセットされる。このUNTAGGED_SWタグエントリーは、好ましくは、ソフトウェアを作成したユーザーによるインストールまたは最初の使用の際に

なされ、指紋処理は、好ましくは、図7で説明するように、タグのないソフトウェアの最初の検出時に、ユーザーの装置104により実行される。

【0301】

図13Aにおいて、確認プログラム(VRP)315が、ステップ414でタグテーブル210にタグのないエントリーを検出すると、ステップ415が実行される。ステップ415の処理では、ステップ410で保護センター103に転送された指紋テーブル126から、各指紋リストを得る。指紋テーブル126は、タグのないソフトウェアのインスタンスそれぞれについての指紋のリストからなる。確認プログラム(VRP)315は、指紋テーブル126中の各指紋リスト X_i と、指紋データ構造体137中のすべての指紋リスト Y_j とを、GCDB300において、前述したように、一般ロケーション指紋チェックを用いて、付き合わせる。指紋リスト X_i と Y_j に指定された数よりも多く一致が見つかり、保護センターは権利侵害のソフトウェアの使用を検出して、処理はステップ420に進み、そこで、報復措置が用意され、呼び出しが実行されたユーザーの装置104に送られる。権利侵害のソフトウェアの権利侵害でないバージョンを作成したソフトウェアのベンダー101にも、通知される。

【0302】

各指紋リスト X_i を保護センター中のすべての指紋リスト Y_j と比較するのは、コンピューターにとって負担が大きく、これは呼び出し中最も負担の大きい処理なので、ある実施形態では、これをいくぶん異なる方法で行う。この実施形態では、反転保護指紋テーブルと呼ばれる指紋リストが作成され、すべての権利侵害のソフトウェアのすべての指紋を含むが、複製された指紋は含まない。この反転保護指紋テーブルを用いて、保護センター103は、各リスト X_i を調べ、リスト中いくつかの指紋が、反転保護指紋テーブル(指紋データ構造体137として保持されている)中の指紋と一致するかを決定する。指定されたのよりも多くの一致が見つかり、各 Y_j に対し X_i の詳細なチェックがなされ、指紋数においてごく近い一致があったか否かの決定がなされる。ステップ415で一致する指紋リストが検出されない場合には、ステップ419の処理に進み、前のステップ411または412で報復措置が規定されていないかを決定する。規定されてい

れば、処理は、前述したように、ステップ420に進む。

【0303】

ステップ419で報復措置が規定されていない場合には、ステップ421に進む。このステップでは、保護センター103に知らされたすべてのタグTAG__INST__SW_nを、ペイパーユース（使用ごとに支払う）タグとして扱う。すなわち、保護センター103は、タグ付きのソフトウェアデータベース138（図9）に、ペイパーユース基準に基づいて課金されるべきソフトウェアのインスタンス111～114すべてのリストを保持できる。ステップ421では、そのようなタグ（第1欄）すべてについて、および1以上のペイパーユースタグの検出について、タグテーブル210を調べ、ステップ421は、保護センター103に、ペイパービューまたはペイパーユース111～114の使用特性に関して、会計情報（図示しない）をソフトウェアベンダー101に向けて送らせる。タグテーブル210中のタグエントリーのRUN COUNTまたはUSE TIME領域は、ペイパーユースの統計の決定に用いることができる。ペイパーユースタグが期限切れになると、TAG__INST__SW_nのためのUSAGE STATUS領域は、“GC_DISABLED”にセットされる。これは、タグに対し使用禁止措置DISABLE（TAG__INST__SW）を用意することでなされる。使用禁止措置は、後述するように、継続メッセージ212に組み込まれる。

【0304】

ステップ421のペイパーユース処理が完了すると、ステップ422が、タグテーブル210中の完全に確認されて期限切れでないタグすべてに対し、継続措置CONTINUE（TAG__INST__SW）を生成する。この継続措置は、継続メッセージ（CM）212に組み込まれる。

【0305】

ステップ423において、確認プログラム315は、継続メッセージ（CM）212を用意し、ユーザーの装置104に返送する。継続メッセージ（CM）212は、いくつかの領域を含んでいる。TIME領域は、クロック304からの現在時刻を示し、ID__TAG__TABLE領域は、呼び出しHASH（EVE

NT_HISTORY) と同時の事象履歴のエンコードと同様に、もとは呼び出し処理のステップ410で保護センター103に送られたタグテーブル210の個別の確認を示す。ACTION領域は、特別のユーザーの装置104の管理プログラム(SP)209に対して利用可能な措置のリストから選択された措置ACTIONS=(ACTIONS1, ACTIONS2, . . . ACTIONS N) のリストを含む。ハッシュ関数値は、措置HASH(ACTIONS) にも含まれ、計算される。最終的に、継続メッセージ212の完全な内容におけるデジタル的に署名された値は、継続メッセージ212が保護センター103になりすますサイトやホストにより偽造されないことを確実にするために、含まれる。好ましくは、署名された値は、次のように表される。

【0306】

SIGN_GC (TIME, ID_TAG_TABLE, HASH (ACTIONS), HASH (EVENT_HISTORY))

【0307】

継続メッセージ(CM) 212の領域のすべてが完成すると、確認プログラム315は、継続メッセージ(CM) 212を、ステップ410で呼び出しを始めたユーザーの装置104内の管理プログラム(SP) 209に、安全に返送または返信する。1つの実施形態では、これには、呼び出し中に装置により与えられた公開のキーを使う。侵害者が同じ公開のキーをもつ2つの装置を設定しても、正しい事象履歴をもつ1つの装置のみが、本発明のこの実施形態にしたがって、継続メッセージ212を処理できる。

【0308】

最終的に、ステップ425において、保護センター103は、呼び出し処理についての呼び出し記録CALL-UP-RECORD_nを作成する。保護センター103は、このTAG_INST_SWに関連するデータ構造体320(図10)に、この呼び出し記録CALL-UP-RECORD_nへのレファランスを付加する。レファランスは、メモリーポインターかCALLUP-RECORDの個別の識別子のいずれかである。呼び出し記録の内容については、図10を参照して前述した。

【0309】

本発明のこの構成の有用性の例は、本発明の特徴のいくつかを明確にする。例えば、ユーザー213がソフトウェアのインスタンス111～114を使用する1年のライセンスを購入し、その1年の期限が切れた後、ユーザー213がライセンスを更新しないとする。ユーザー213が更新しないので、ソフトウェアベンダー101は、ユーザー213がもはやライセンスを保持していないソフトウェアのインスタンス111～114を使用禁止にすることを望む。本発明によれば、ベンダー101は、ソフトウェアのインスタンス111～114に関連する保護センター103にポリシーPOLICY (TAG__INST__SW) を単に設定して、インスタンス111～114が備えられた装置104から保護センター103への次の呼び出しにおいて、インスタンスを使用禁止にできる。このようにして、強力な使用管理が、ユーザー213にソフトウェアのインスタンス111～114の返却を求めることなしに、なされる。ユーザー213が、後でライセンスの更新を希望するならば、ベンダー101は、ただ保護センター103でポリシーPOLICY (TAG__INST__SW) を変更すればよく、そうすると、次の呼び出しが、ユーザーの装置104のタグテーブル210を、インスタンス111～114について“CONTINUED”ステータスタグTAG__INST__SWとして更新する。

【0310】

保護センター103に用意された継続メッセージCM212の種々の要素、および、CM212に組み込まれた前述のデジタル署名は、本発明の実施形態でいくつかの重要な目的を果たす。継続メッセージ212は、それを受けるユーザーの装置104の管理プログラム209に、装置のタグテーブル210のUSAGE STATUS欄をどう更新するか、および、もしあるなら、どの報復措置をとるか、について指示する。識別ハッシュ関数およびCM212 (図13Bの423) の他の値は、要求された呼び出し処理を完遂するために、不正なユーザー213が、ユーザーの装置104 (すなわち、104～107のうちの1つ) からの現在の呼び出しに応じて保護センター103により実際に与えられた1つ以外のあらゆる継続メッセージ212を使用することを、仮想的に不可能にする。

また、敵対者や敵対ホストは、不正CM212を装置104に送ることによって、ユーザーの装置（すなわち、104）へのサービスを拒否するような損害を与えることはできない。

【0311】

前記の好ましい実施形態で述べたように、本発明は、ソフトウェアベンダー101により作成され、配布された（すなわち、販売された）ソフトウェアのインスタンス111～114、または、ユーザーの装置104によるアクセスで権利侵害され、不正に配布されたインスタンスの使用を、検出し、制御し、かつ管理する機構を提供する。各ソフトウェアのインスタンス111～114を個別に識別する、偽造できず、認証されたタグTAG__INST__SWを与えることにより、使用管理がなされる。好ましい実施形態では、TAG__INST__SWが適正にソフトウェアのインスタンスINST__SWに関連しているかを確認するのに、指紋位置の同一性（same location fingerprinting）を用いる。

【0312】

指紋は、やや異なる目的にも同様に用いることができる。そのような1つの目的は、オペレーティングシステム207のテキストの完全性（textual integrity）のチェックである。これは、プログラムの一部に、他の部分または他のプログラムを前記指紋処理でチェックさせることにより、なしえる。これは、例えば管理プログラム209やオペレーティングシステム207の不法変更を防止する。他の実施形態では、電子的にプログラム可能なリードオンリーメモリーのような外部ハードウェアが、機器または装置104～107の電源が入っているときに、このチェックを行える。いずれの場合も、チェックプログラムは、前述のように、オペレーティングシステムプログラム207のいくつかの部分でハッシュ指紋を計算でき、例えば、指紋の不一致を見つけると、装置の失敗を起こす。また、指紋は、オペレーティングシステム207により管理プログラム209テキストをチェックするのにも、用いることができる。同様に、管理プログラム209は、事象履歴のハッシュを確認または認証のチェックに用いることができる。

【0313】

これは、例えば、以下のように動作する。管理プログラム209は、MD5の

ような増大ハッシュ関数法を用いて、タグテーブル210のデータのハッシュを各更新の後に更新できる。タグテーブル210を新しい事象で更新する前に、定期的に、管理プログラム209は、保持するハッシュ関数値がタグテーブル210のハッシュと等しいことを確認できる。これらのチェックのいずれかでも失敗すると、管理プログラム209またはオペレーティングシステム207は、報復措置をとることができる。このように、本発明の構成は、本発明自体として動作するソフトウェアを不法変更する装置またはソフトウェアを検出するために、用いることができる。

【0314】

指紋のさらなる用途は、ソフトウェアのインスタンス111～114のタグ120に対する要求に応じてタグサーバー102に出された特定のベンダーソフトウェアが、他の適法なベンダーのソフトウェアSWからの不法な複製または派生物でないことの確認である。そのような行為が可能とすると、侵害者であるベンダーが、認証されたタグ120を供給したタグサーバー上の、他の適法なベンダーのソフトウェアSWを配布するのを許すことになる。本発明のこの構成は、あらたに生成したソフトウェアを指紋付けし、一般位置指紋（general location fingerprinting）を用いて、新たなソフトウェアを既存のソフトウェアと比較し、新たに出されたベンダーソフトウェアが適法なベンダーソフトウェアSWに酷似していないかを調べて、このような形態の侵害を防止する。

【0315】

ソフトウェアのインスタンス111～114は、それがインストールされるときまたは最初に使用されるときにチェックされたタグをもっている。タグは、また、後にもチェックされる（すなわち、ハッシュ、署名または呼び出し処理を介して）。ソフトウェアが最初に使用されるまで待つ理由の1つは、ソフトウェアが大容量の場合、最初にインストールされるときよりも、ソフトウェアの実行中の方が、チェックの負担が少ないからである。

【0316】

失敗を理由に、装置の状態（ステータス）は、前の状態に再構築される。この場合、ユーザー213は、古いHEADER_TAG_TABLEを送ってもら

う必要があることを通知するために、保護センター103に連絡しなければならない。この特典を受けるにつき疑わしいユーザーは、保護センター103に容易に追跡される。

【0317】

図14は、本発明の他の実施形態で用いられるデータ構造体を示し、この実施形態によれば、保護センター103が共有データファイルを提供するソフトウェアを呼び出すことを不要にできる。ワープロのプログラムを例にとる。ワープロのファイルと同様に、ワープロのソフトウェアも、知人間でよく交換される。一般的には、最初のケースは許されるが、2番目のソフトウェアアプリケーション交換のケースは許されない。このような権利侵害を防止するために、本発明の実施形態は、図14のデータ構造体600に示すように、例えばID_TAG_TABLE同様に、プログラムに付随するTAG_INST_SW120を書き込み、各共有ファイルの不明な位置 (invisible location) への最近のアクセス時刻を書き込んで、ソフトウェアアプリケーションプログラムを変更できる。また、同図に示すように、プログラムは、TAG_INST_SWおよびTAG_TABLE601への最近のアクセス時刻を書き込む。

【0318】

共有データファイル (すなわち、例えばドキュメントで、ここではSSDで参照される) 中の不明な位置 (すなわち、ユーザーにとって不明な) に蓄積されたデータ構造体600は、共有ソフトウェアデータSSDファイルのコメントセクションに置かれ、好ましくは3つの独立変数を用いる偽造できないハッシュ関数に付随できる。

【0319】

図15は、前述のソフトウェア権利侵害保護機構を提供する本発明の実施形態のステップを示す。図15のステップ700において、ID_TAG_TABLE Xをもつ第1のユーザーの装置 (すなわち、ユーザーの装置104) の管理プログラムSP 209が、共有ソフトウェアデータSSDへのアクセスを検出すると、管理プログラム209は、共有ソフトウェアデータSSDと、ある特定の時刻にTAG_INST_SW Tをもつソフトウェアのインスタンス (すな

わち、111～114のうちの1つ)によりアクセスされた共有ソフトウェアデータSSD内の所定の位置の記録とを調べる。そして、ステップ701で、ソフトウェアのインスタンス(潜在的に他の機械または他のユーザーの装置(例えば105)にある)が、共有ソフトウェアデータファイルSSDを実行し、アクセスしようとする、ユーザーの装置105の管理プログラム209は、共有ソフトウェアデータファイルSSD中にデータ構造体600があることを検出し、SSDからタグTを得て、ユーザーの装置105(共有ファイルを得ている装置、ただし、ファイルSSDの装置を作製する必要はない)のタグテーブル210をチェックし、タグテーブル210にタグTがあるかを調べる。タグTがない場合は、共有ソフトウェアデータSSDにアクセスするために第2の装置105(共有データを得ている装置)で使用されたソフトウェアのインスタンスは、複製されたものではなく、アクセスが認められてステップ703の処理に進む。

【0320】

一方、ステップ701で、共有ソフトウェアデータSSD中のデータ構造体600にタグTがある場合には、処理はステップ702に進む。ステップ702では、第2の装置105の管理プログラム209は、タグTの付いたソフトウェアのインスタンス(例えば、第2の装置105のインスタンス111～114のうちの1つ)が、SSDに組み込まれたデータ構造体600に示された時刻に共有データファイルを書き込んだかを調べる。書き込んでいなかった場合には、権利侵害が起こったのであり、管理プログラム209は、ステップ704で第2のユーザーの装置に対し報復措置を実行する。ステップ702で、第2の装置105の現在のソフトウェアのインスタンス111～114が、SSDに組み込まれたデータ構造体600中の情報により示されたように、共有ソフトウェアデータSSDにアクセスすることが決定されると、処理は703に進み、そこで、共有ソフトウェアデータへのアクセスが認められる。

【0321】

本発明の他の実施形態では、同じソフトウェアの異なるソフトウェアのインスタンスは、装置識別子によって異なる。このような実施形態の利点は、保護センターとの連絡の必要性が低減されることである。不利な点は、各ソフトウェアの

インスタンスは、異なっていなければならない（タグの有無まで対比して）、装置から装置へ移せないことである。この実施形態では、装置識別子は、利用できる場合にはプロセッサ識別子から作成されるか（インテル社製のペンティアム（登録商標）IIIのようなプロセッサは、プロセッサ識別子をもっている）、好ましくは管理プログラム識別子から作成され、前述したようなプロセッサ識別子を組み込んでいる。各ソフトウェアのインスタンスは、テストでそのソフトウェアのインスタンスを使用する装置の識別子を、ソフトウェアのインスタンスのコードに組み込む。そのようなテストは、C言語で、例えば“if文”として、表される。テストでは、組み込まれた識別子を装置識別子と比較する。ソフトウェアは、実行中、テストを行う。比較が成功（一致）すると、装置はソフトウェアのインスタンスを使用する。比較が失敗（不一致）であると、装置はインスタンスを使用せず、管理プログラムに報復措置をとるよう通知する。侵害未遂は、装置識別子をチェックしないように、プログラムを変更する。これは、タグ付きのソフトウェアを、あたかもタグなしで権利侵害であるかのようにするのに似ている。装置テストが変更され、または除去されたソフトウェアは、図13Aに詳述し、ステップ414から始まる指紋に基づく機構により検出される。

【0322】

この実施形態の変形では、ベンダーが、装置識別子と、装置識別子を組み込むソフトウェアのインスタンスのハッシュのデジタル署名との両方を送る。これは、次のように計算される。

【0323】

`SIGN_VENDOR (HASH_INST_SW)`

ただし、`HASH_INST_SW=HASH (SW, DEVICE_IDENTIFIER)`

【0324】

ここで、`SIGN_VENDOR`は、ベンダーのデジタル署名であり、`HASH_INST_SW`は、ソフトウェアの内容（すべてのインスタンスに個別である）に組み込まれた`DEVICE_IDENTIFIER`を加えて計算される。装置識別子を組み込んだソフトウェアのインスタンスは、好ましくは、その識別

子を、ソフトウェアの内容の最初または最後に配置し、ハッシング処理の負担を軽減する。第2のテストでは、デジタル署名SIGN_VENDORが確認され、第3のテストでは、送られたHASH_INST_SWが、ソフトウェアのインスタンスのハッシングの結果値に等しいかを確認する。これらのテストは、ユーザーの装置の管理プログラムで実行される。デジタル署名が認証されないか、HASH_INST_SWが、受けたソフトウェアのインスタンスのハッシュと異なる値をもつと、管理プログラムにより報復措置がとられる。

【0325】

以上においては、タグサーバー102、保護センター103およびベンダー101は、分離されたものとして説明した。他の実施形態では、これらの役割をまとめてもよい。例えば、1つのサイト、ネットワークホストまたはサーバーは、保護センター103およびタグサーバー102の両方として機能できる。あるいは、ソフトウェアベンダー101は、3つすべての役割を果たすことができる。さらにまた、各処理または役割を分離しても、前記実施形態における1つの要素（すなわち、タグサーバー、保護サーバー、ベンダー）に割り当てたいくつかの機能は、他の要素により実行できる。例えば、同一位置指紋（same-location fingerprinting）は、タグサーバー102に代わって、ベンダー101で実行できる。

【0326】

本発明を好ましい実施形態により図示し、説明してきたが、当業者には、前記特許請求の範囲で定まる本発明の精神と範囲から逸脱することなく、形状または細部の各種の変更が実行可能であることは理解されるであろう。

【図面の簡単な説明】

【図1】

本発明の一実施形態かかる情報システムの図である。

【図2】

本発明の一実施形態かかるシステムにおける情報の流れの詳細図である。

【図3A】

本発明の一実施形態に従ってソフトウェアインスタンスに対して署名されたタ

グを作る処理ステップを示すフローチャートである。

【図3B】

本発明の一実施形態に従ってソフトウェアインスタンスに対して署名されていないタグを作る処理ステップを示すフローチャートである。

【図3C】

本発明の一実施形態に従ってソフトウェアインスタンスに対して指紋を持つ署名されていないタグを作る処理ステップを示すフローチャートである。

【図4】

本発明の一実施形態にかかるユーザー装置のアーキテクチャーを示す図である。

【図5】

本発明の一実施形態に従ってユーザー装置にベンダーソフトウェアをインストールする処理ステップを示すフローチャートである。

【図6】

本発明の一実施形態にかかるタグテーブルの内容を示す図である。

【図7】

本発明の一実施形態に従ってユーザー装置にタグのないソフトウェアをインストールする処理ステップを示すフローチャートである。

【図8】

本発明の一実施形態に従ってソフトウェアの使用管理を実施する本発明のシステムによって実行される高レベル処理ステップを示すフローチャートである。

【図9】

本発明の一実施形態にかかる保護センターのアーキテクチャーを示す図である。

【図10】

本発明の一実施形態にかかるソフトウェアインスタンスに対する保護センター記録の内容を示す図である。

【図11】

図11は、ベンダーのソフトウェアに関するベンダーの権利を侵害するソフト

ウェアをベンダーが検出した際に、本発明の一実施形態にかかる保護センターによって実行される処理のフローチャートである。

【図12】

本発明の一実施形態に従って保護センターに対する呼び出し手順を実行する際に、ユーザー装置の管理プログラムによって実行される処理ステップのフローチャートである。

【図13A】

本発明の一実施形態に従って実行される保護センター呼び出し処理ステップのフローチャートである。

【図13B】

図13Aの続きのフローチャートである。

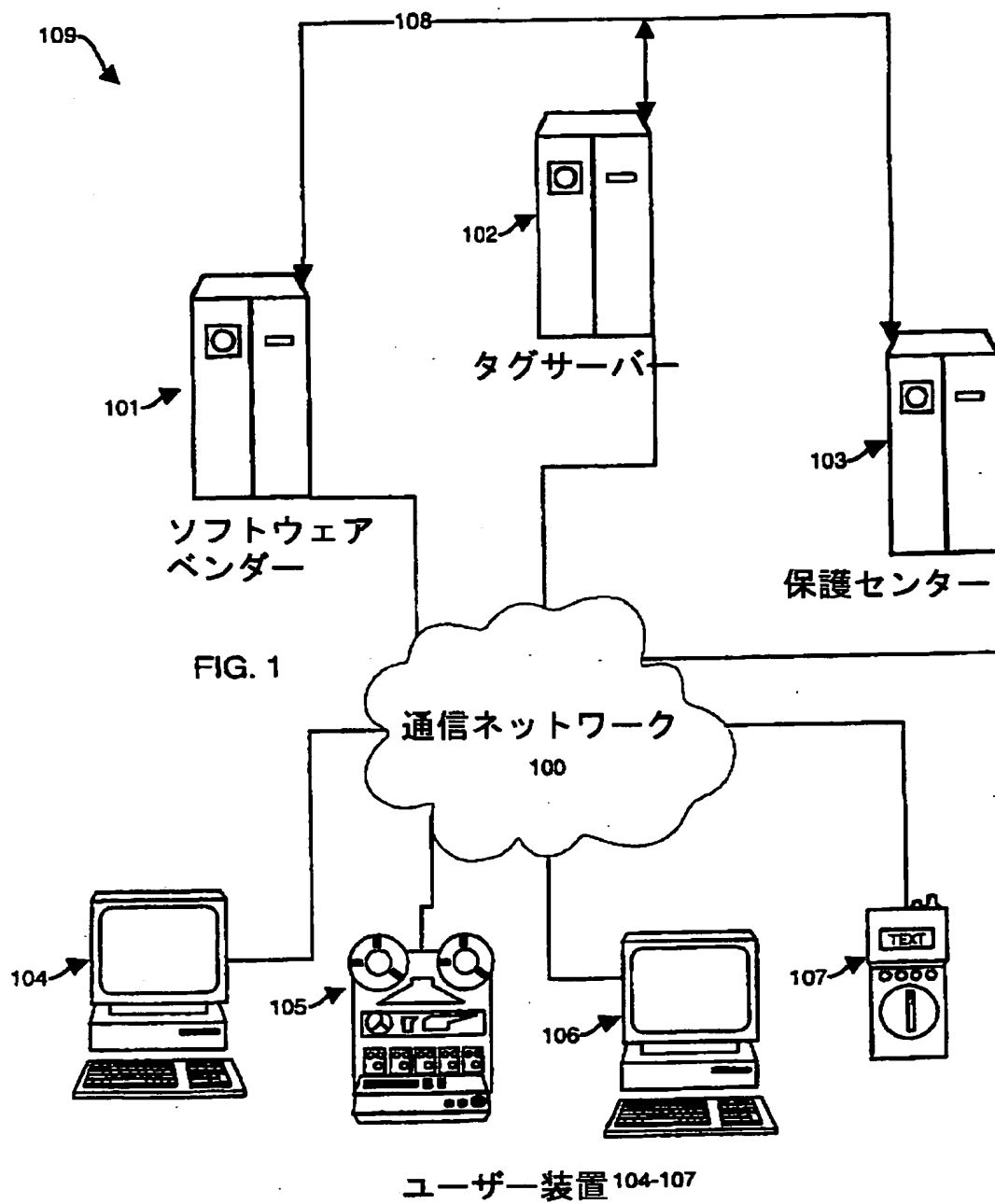
【図14】

保護センターの呼び出しなしに、本発明の一実施形態で使用されるデータ構造体を示す図である。

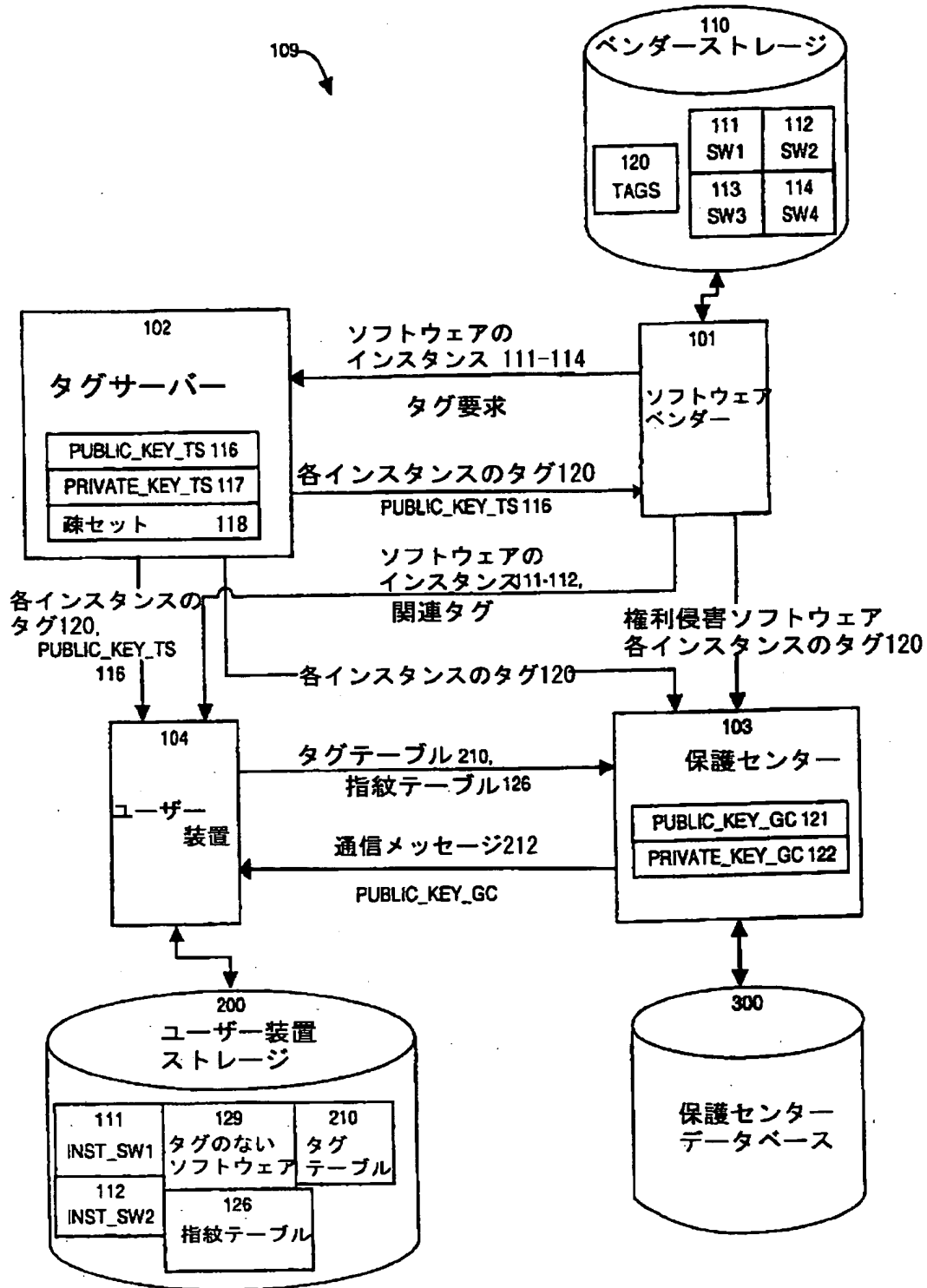
【図15】

保護センターの呼び出しなしに、本発明の一実施形態におけるユーザー装置の管理プログラムによって実行される処理ステップのフローチャートである。

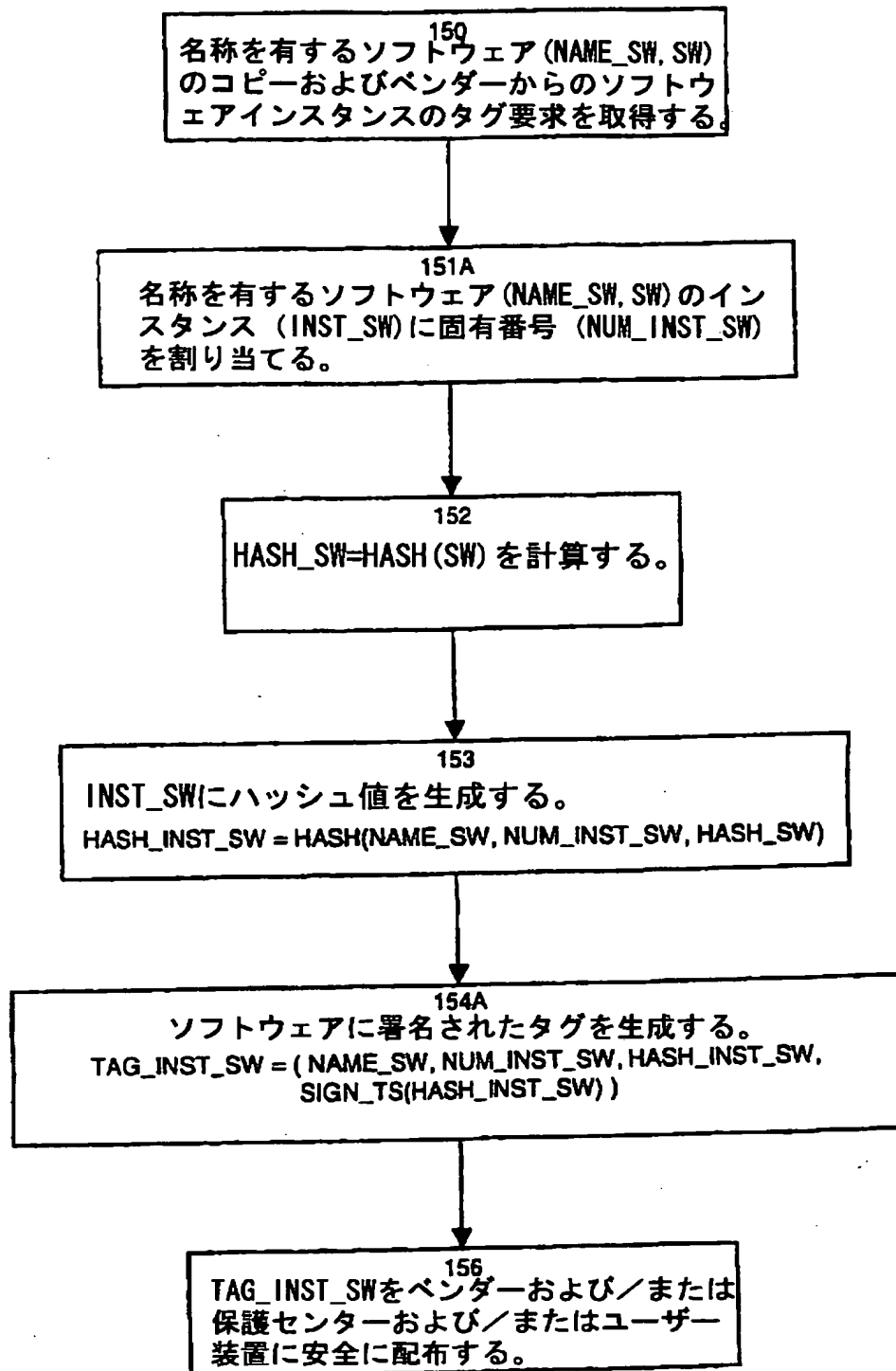
【図1】



【図2】

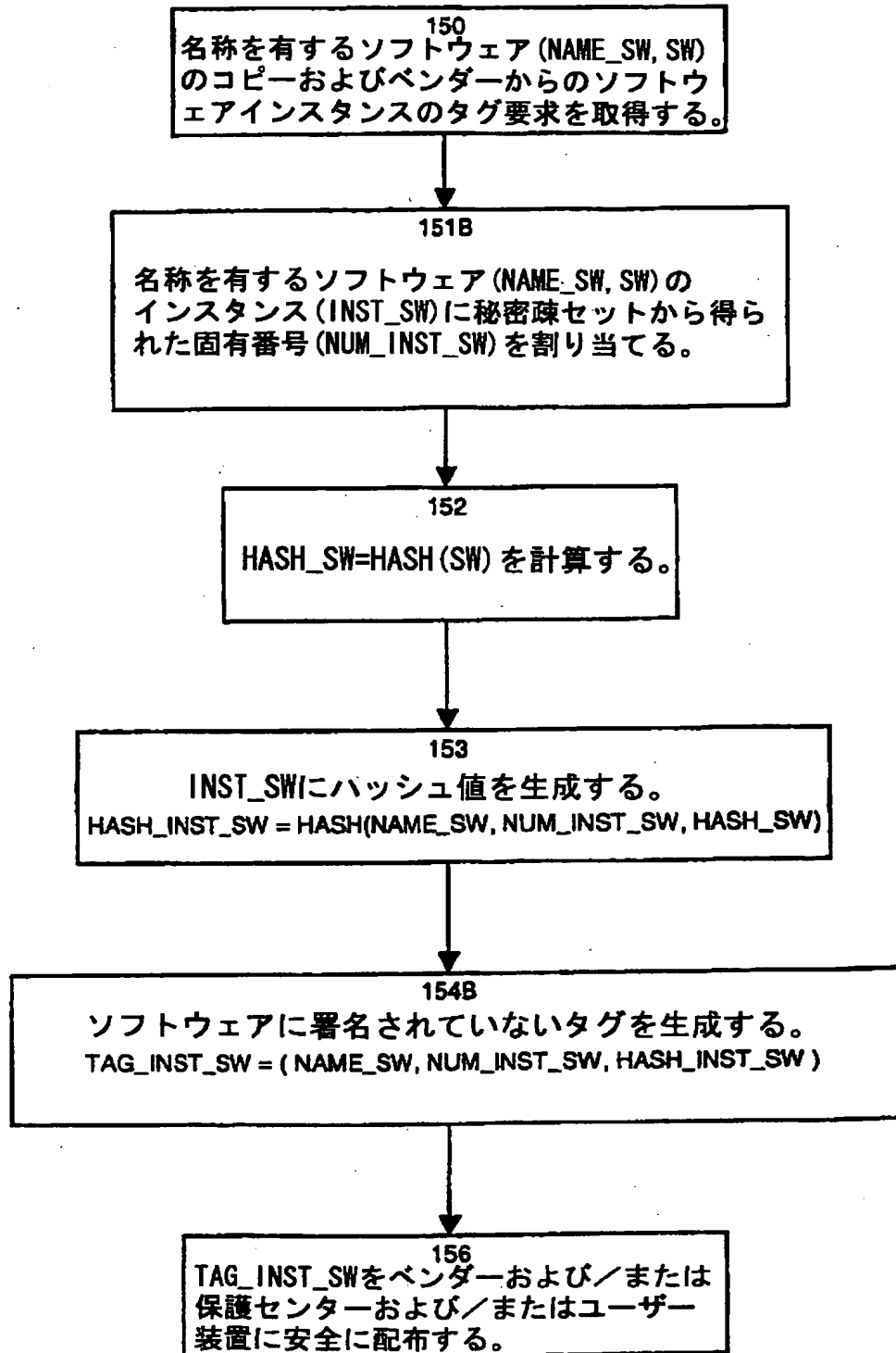


【図3A】



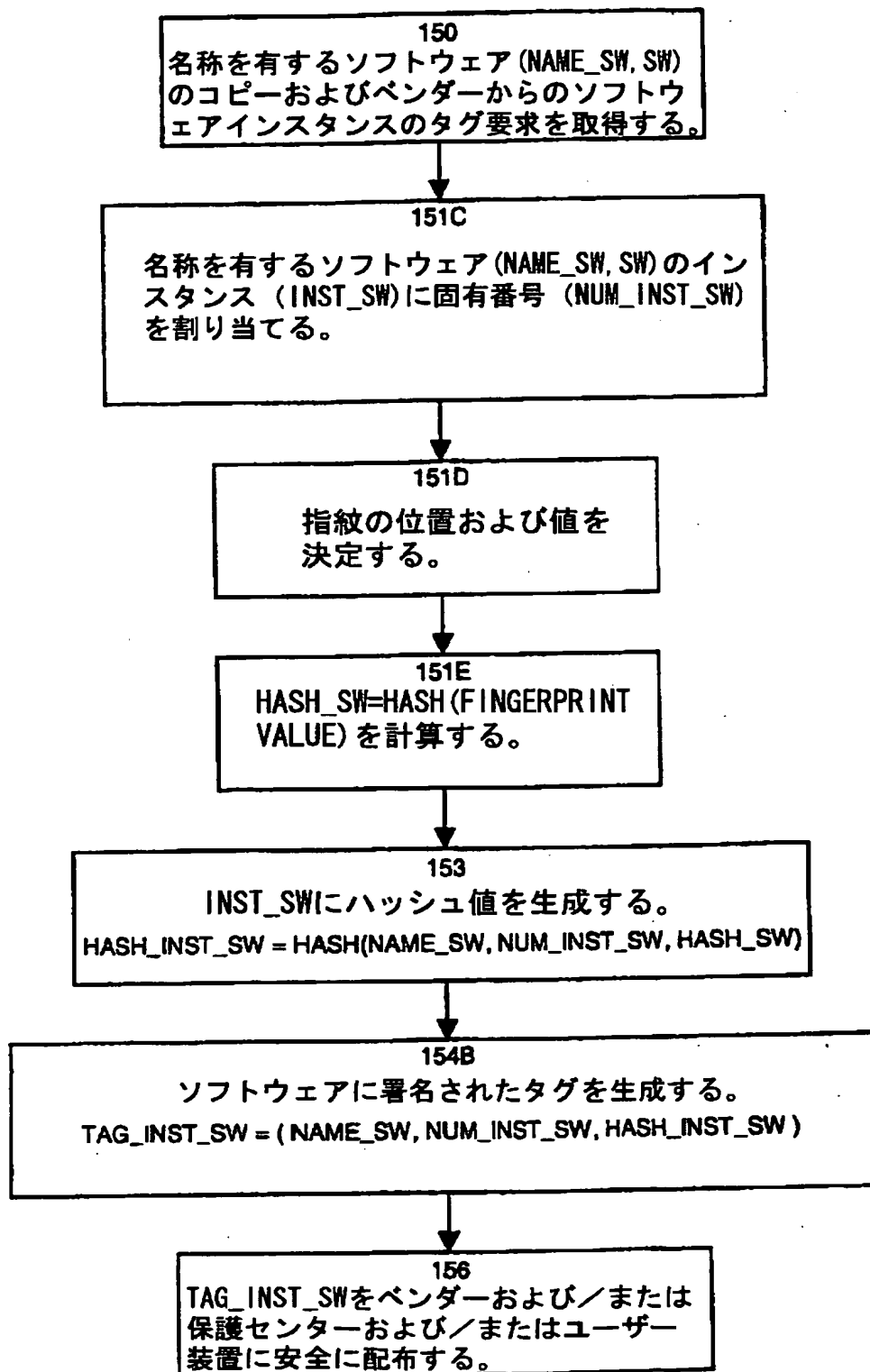
タグサーバーによる署名されたタグの生成

【図3B】



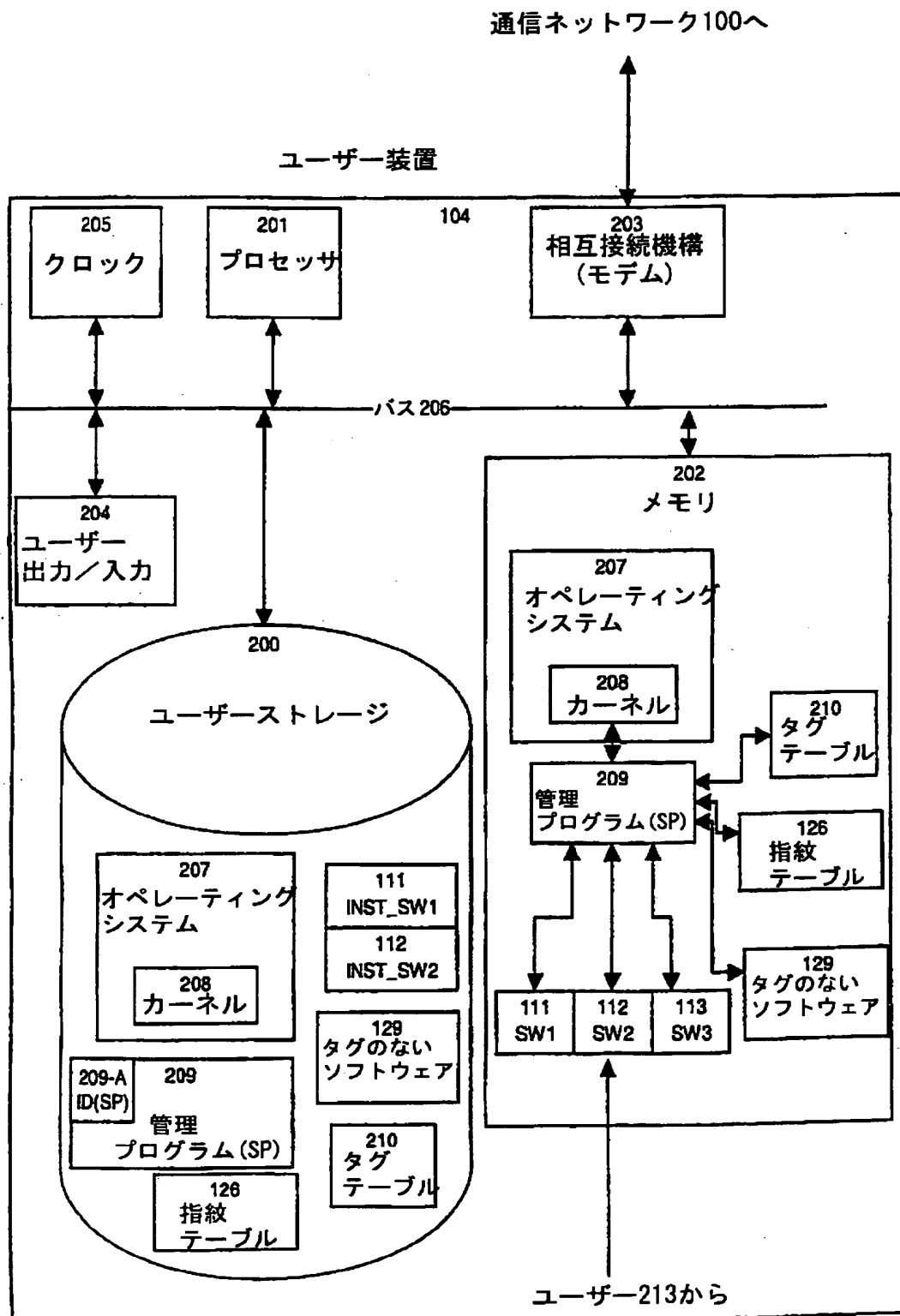
タグサーバーによる署名されていないタグの生成

【図3C】

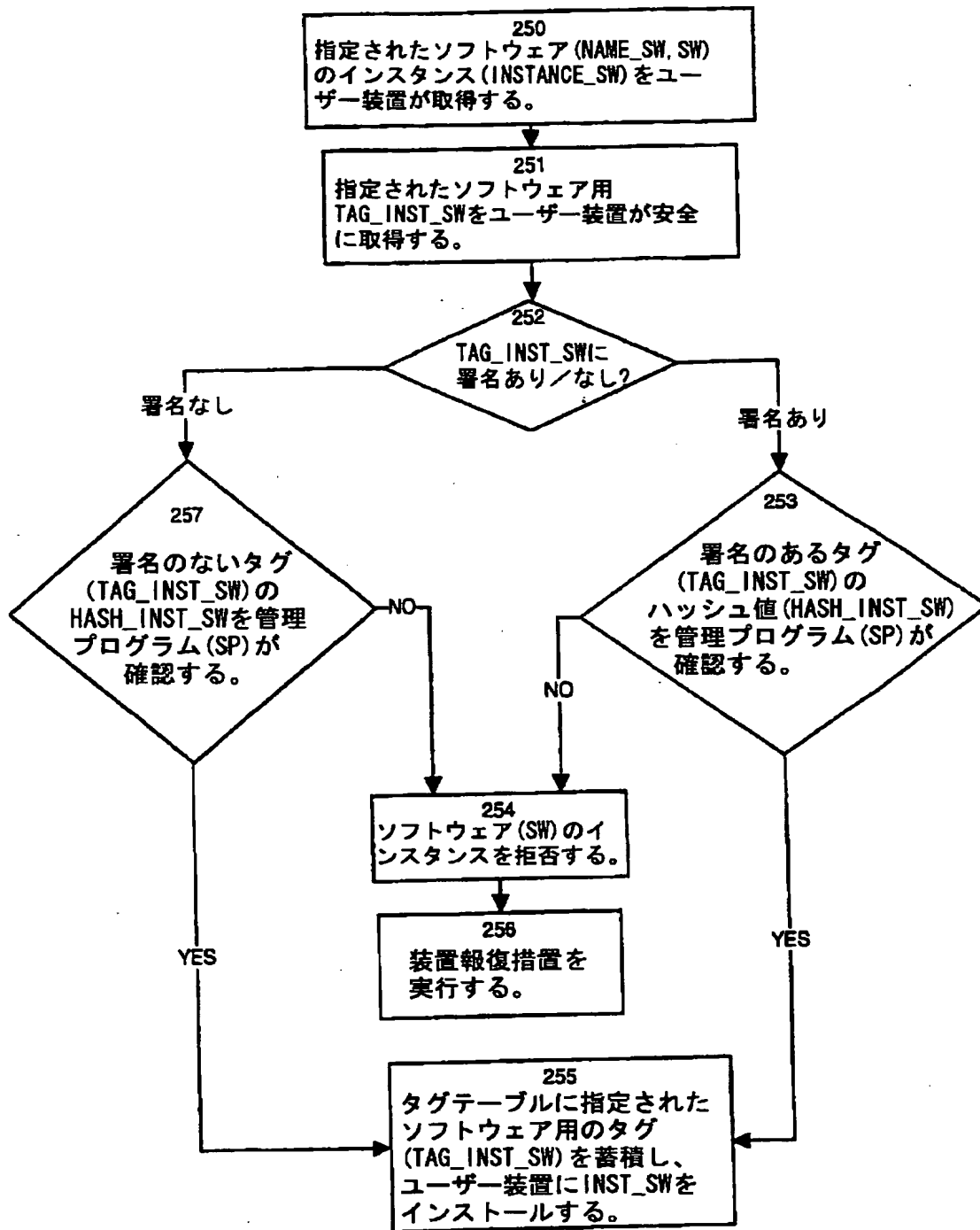


タグサーバーによる同一位置指紋を用いたタグ生成

【図4】



【図5】



ベンダーソフトウェア (SW) のタグのあるインスタンス
をユーザー装置にインストール

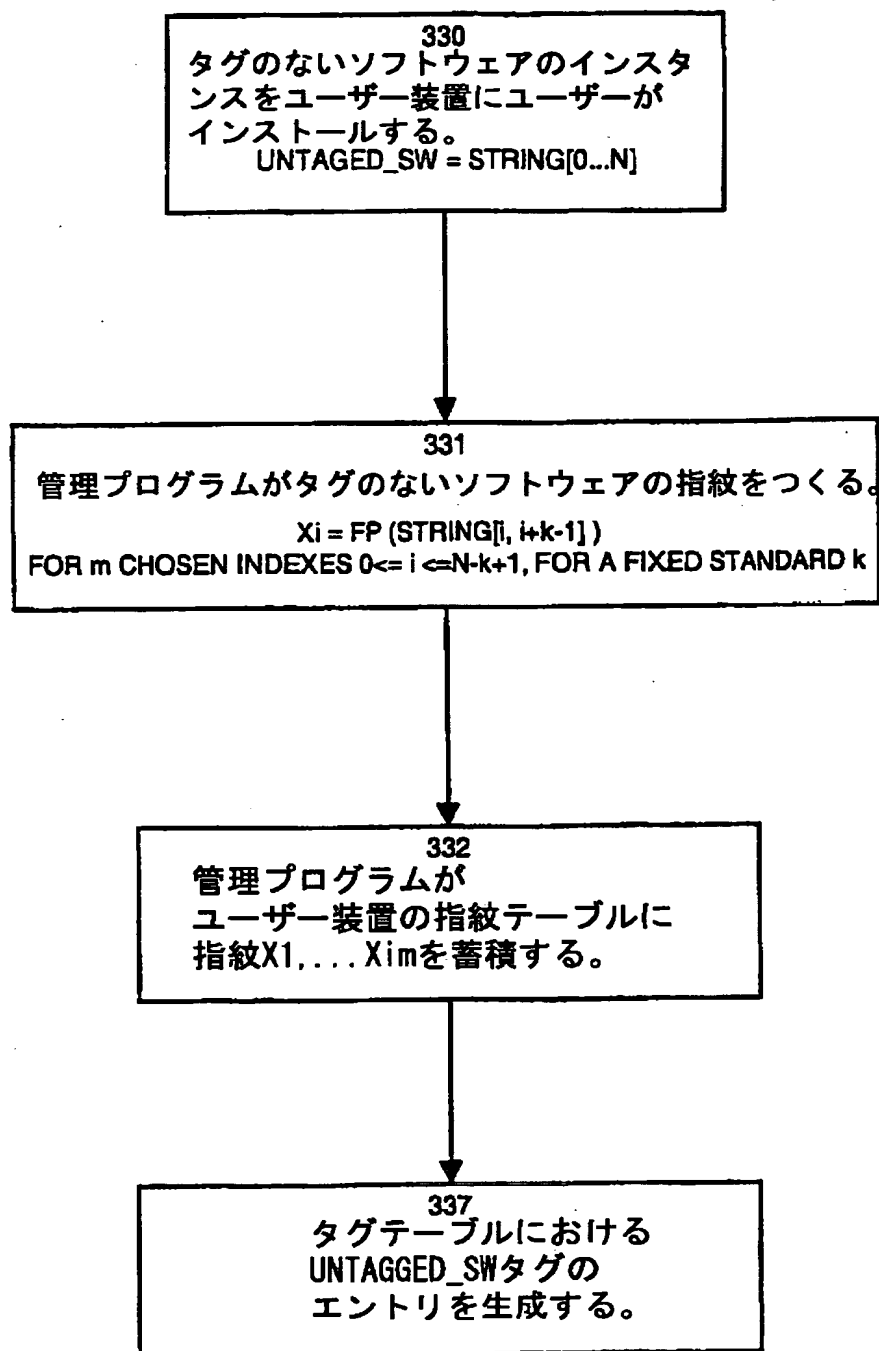
タグテーブル

210					
HEADER_TAG_TABLE=(ID_TAG_TABLE, LAST_GC_CM, LAST_CALLUP_TIME, NUMBER_DEVICE_BOOTUPS)					
タグ	使用状況	動作時刻	ラン カウント	使用時間	
TAG_INST_SW1	CONTINUED	11/22/98 11:02	10	00:58	
TAG_INST_SW2	INSTALLED REMOVED	11/24/98 21:36 11/26/98 09:31	3	00:11	
TAG_INST_SW3	GC_DISABLED	11/22/98 11:02	0	00:00	
TAG_INST_SW4	CONTINUED	11/25/98 14:17	4	10:34	
UNTAGGED_SW1	UNTAGGED	11/25/98 14:17	0	00:00	

注: ID_TAG_TABLE = (ID(USER)*, ID(DEVICE)*, ID(PP)*, ID(OS)*)

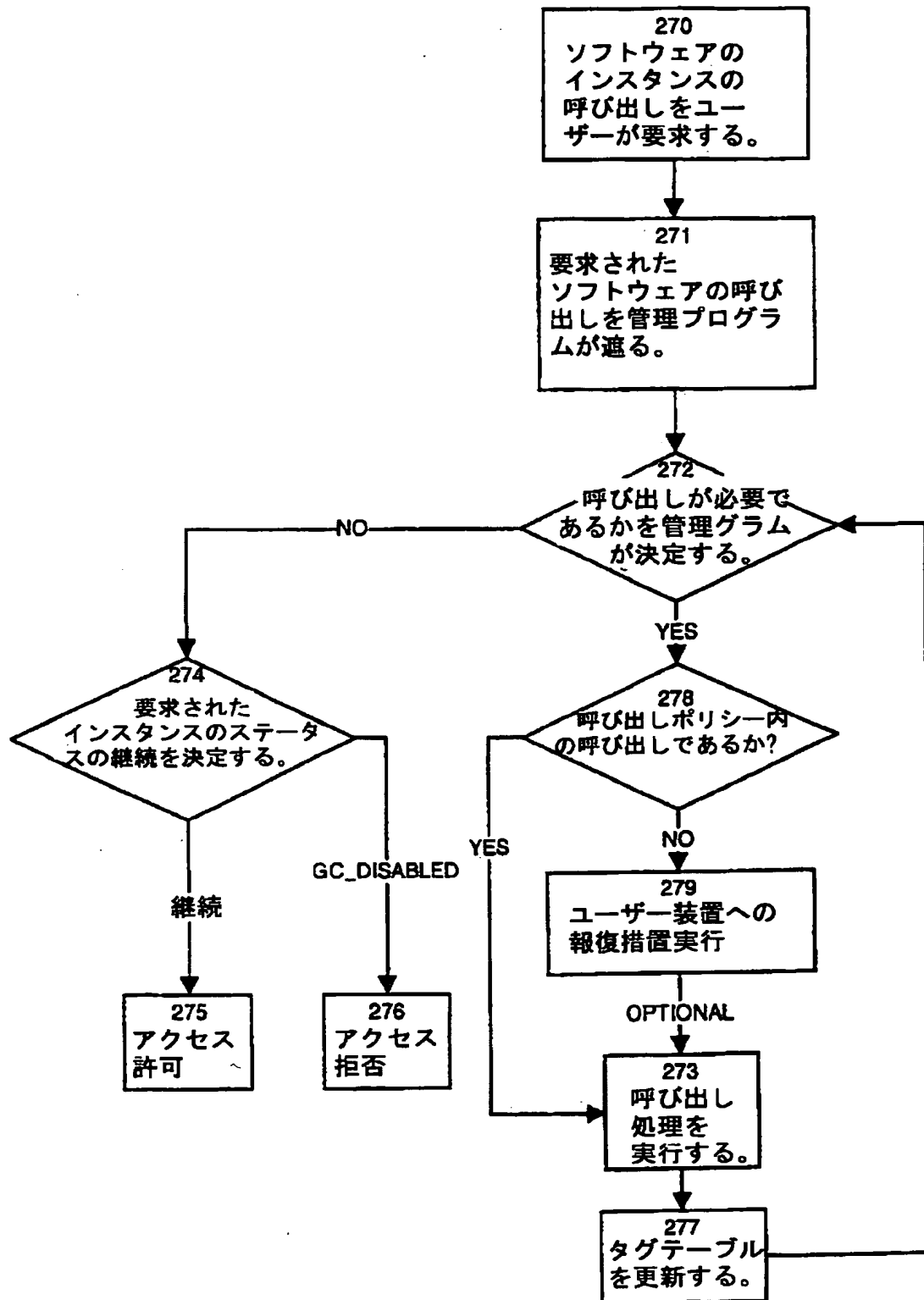
・不明の場合もある。

【図7】



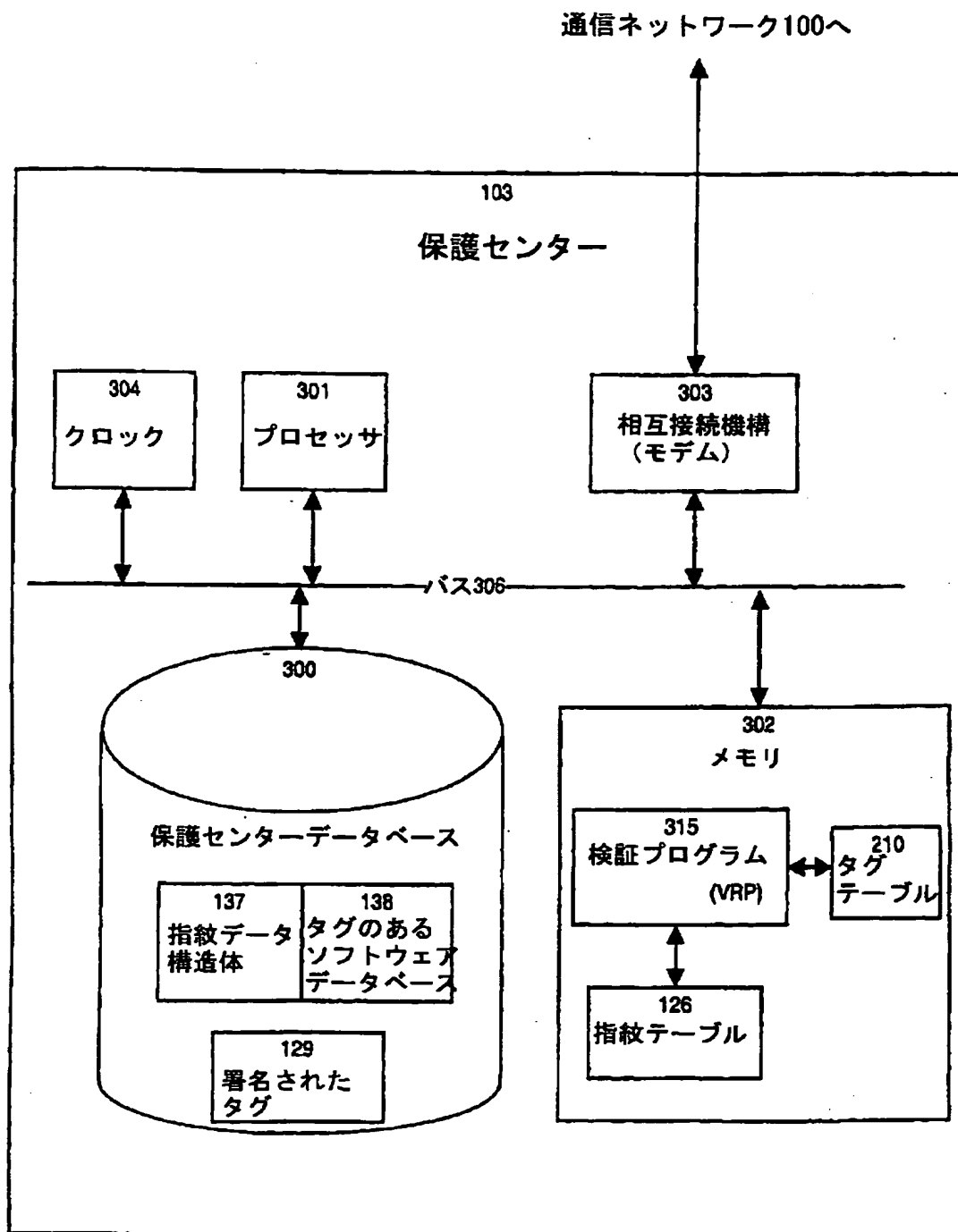
タグのないソフトウェア (UNTAGED_SW) を
ユーザー装置にインストール

【図8】



使用管理手順

【図9】

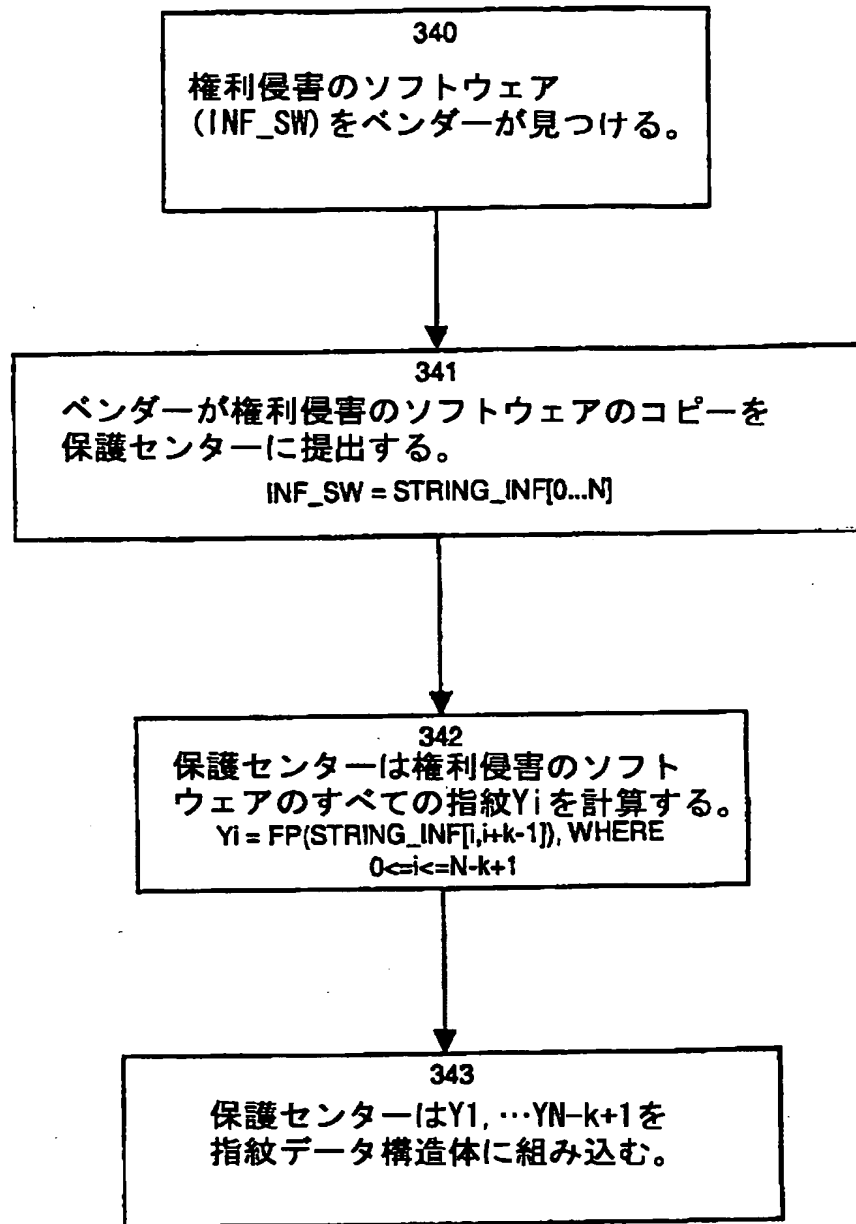


【図10】

320	(NUM_INST_SW, TAG_INST_SW, NAME_SW, HASH(SW), POLICY(TAG_INST_SW), POINTER TO SPARSE_SET(SW)) --> CALLUP_RECORD1--> CALLUP_RECORD2 --> CALLUP_RECORD3 --> ..
321	CALLUP_RECORD-N = (CALLUP_TIME, HEADER_TAG_TABLE OF CALLING DEVICE, LAST_CALLUP_TIME FROM DEVICE, HASH(TAG_TABLE), ACTIONS)

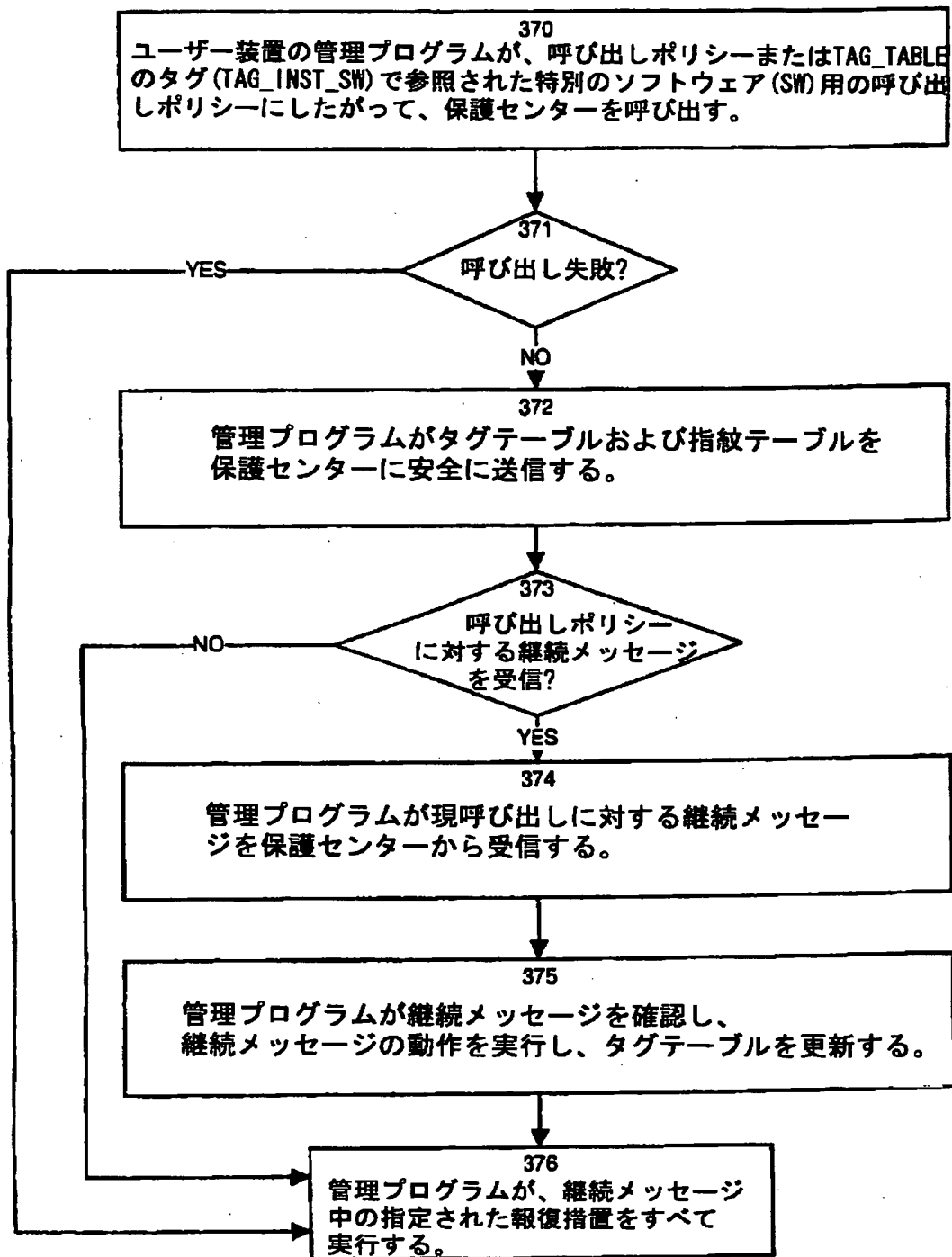
保護センタ－データ

【図11】



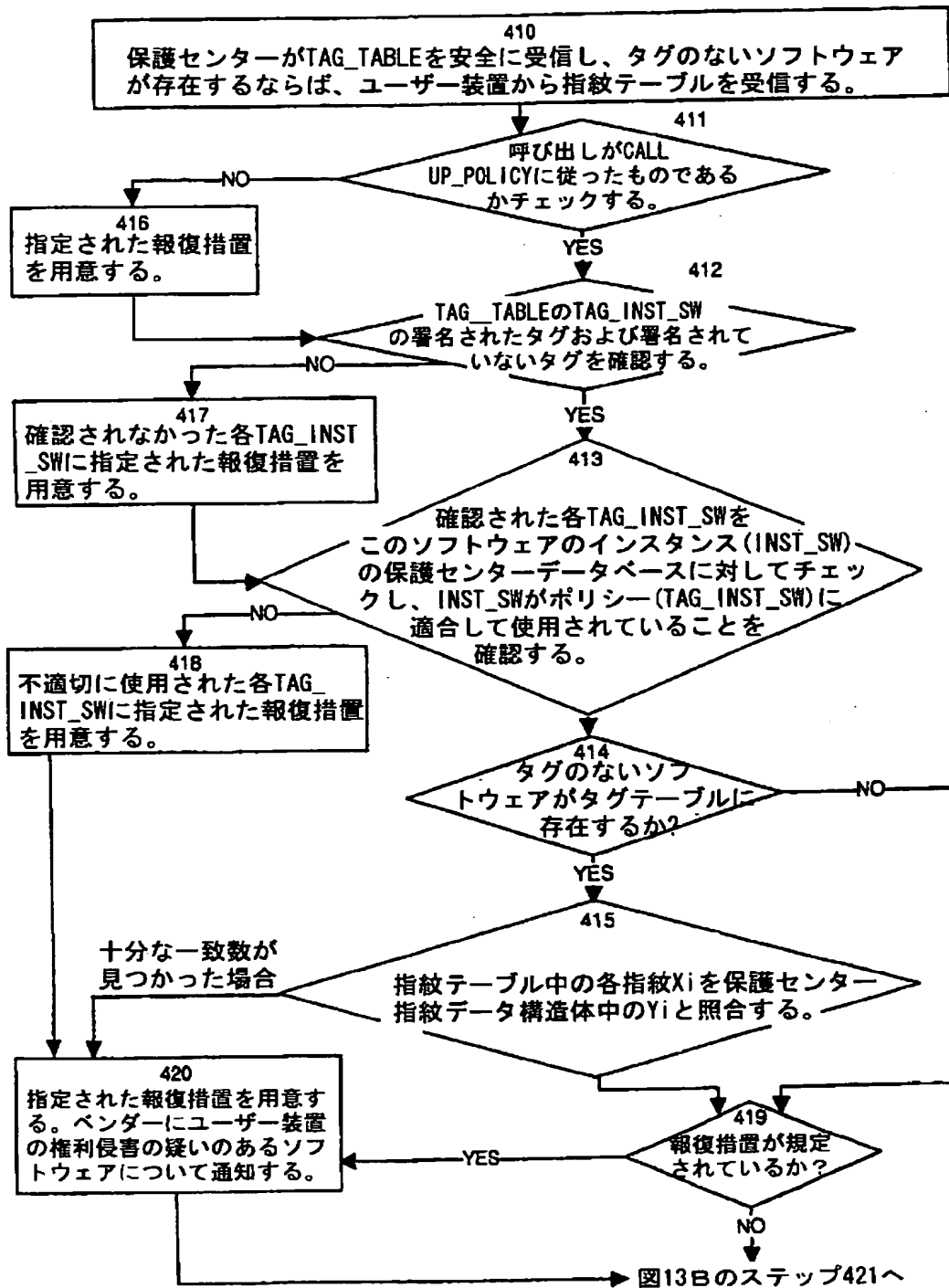
保護センターが権利侵害のソフトウェアを認識

【図12】



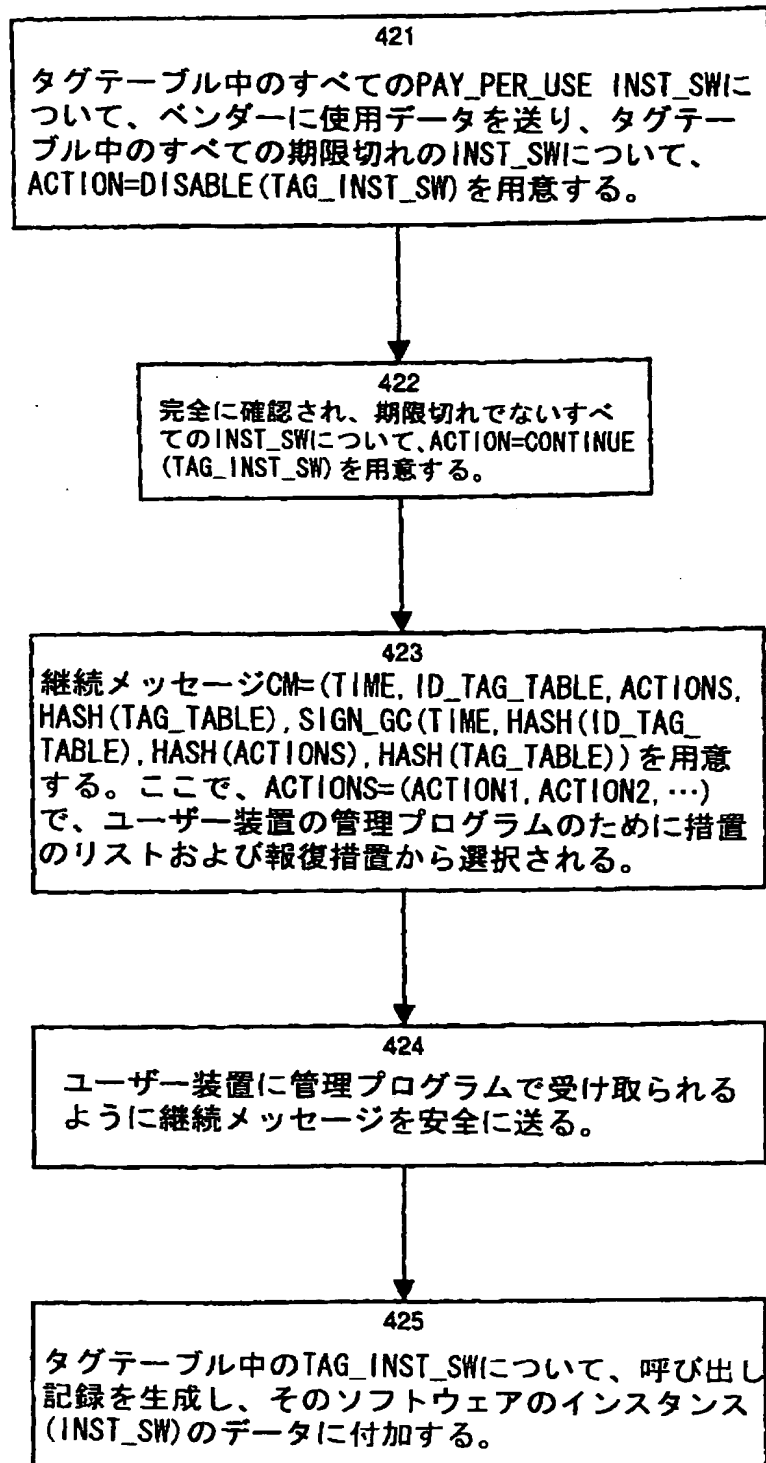
ユーザー装置から保護センターの呼び出し

【図13A】



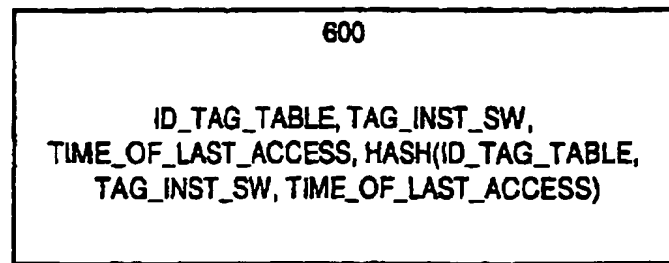
保護センター呼び出し処理

【図13B】



保護センターに呼び出し処理

【図14】



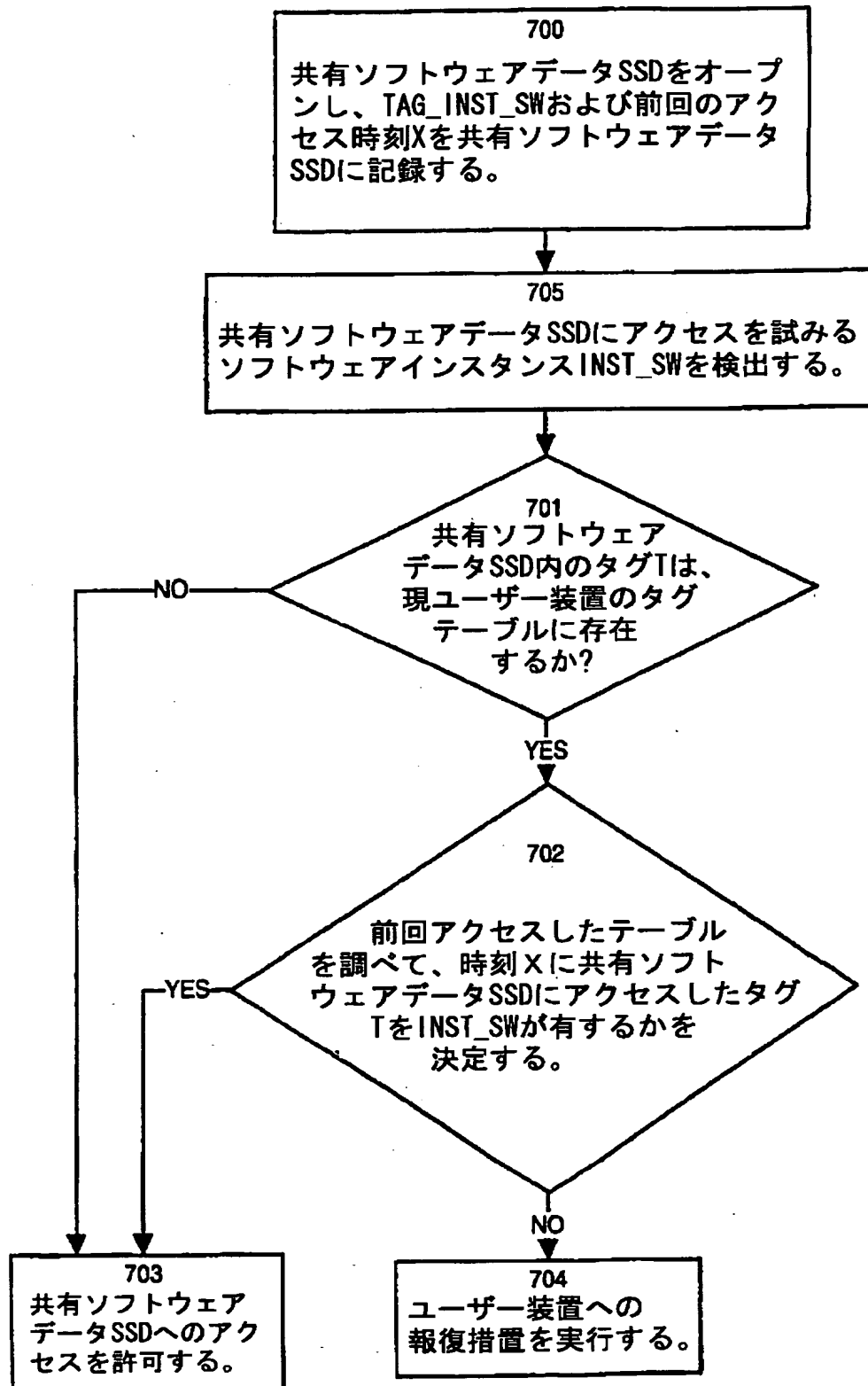
共有ソフトウェアデータと共に伝送されたデータ

601

タグ	アクセス時刻
TAG_INST_SW1	1999年 2月2日 午前 09:11 1999年12月2日 午前 07:24 1999年 2月1日 午後 09:36 1999年 2月2日 午前 11:17...
TAG_INST_SW2	1999年 1月 5日 午後 10:19 1999年 2月 8日 午前 08:34 1999年 2月 5日 午後 01:33...
TAG_INST_SW3	1999年 1月12日 午前 06:41

共有ソフトウェアデータSSDアクセステーブル

【図15】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/US 00/11821

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EP0-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	W0 98 45768 A (NORTHERN TELECOM LTD) 15 October 1998 (1998-10-15) abstract; figures 1,3C,5 page 4, line 18 -page 5, line 32 page 6, line 38 - line 37 page 8, line 17 -page 10, line 24 page 11, line 7 -page 12, line 2 page 18, line 33 -page 19, line 18 page 20, line 26 -page 23, line 21 page 24, line 30 -page 25, line 26	57,58, 91,92, 111-114
Y	---	1-5, 7-12,59, 62-65, 93,96
	--- --	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search

10 October 2000

Date of mailing of the international search report

20.02.2001

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel: (+31-70) 340-2010, Tx: 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Sigolo, A

INTERNATIONAL SEARCH REPORT

Internal Application No
PCT/US 00/11821

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 825 883 A (ARCHIBALD WILLIAM CHARLES ET AL) 20 October 1998 (1998-10-20) abstract; figures 2-4,6 column 4, line 1 -column 6, line 40 column 7, line 32 -column 9, line 67 column 10, line 57 -column 11, line 51 column 18, line 44 -column 19, line 39 column 28, line 41 - line 55	53-55, 57-59
Y		1-5, 7-12,59, 62-65, 93,96
A	<p style="text-align: center;">---</p> US 5 823 987 A (JOHNSON HERRICK J ET AL) 11 June 1991 (1991-06-11) abstract; figure 1 column 2, line 49 -column 5, line 42 <p style="text-align: center;">-----</p>	13-18

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 00/11821**Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)**

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-18, 53-70, 91-93, 96-102, 111-114

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest.
- ☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-18, 53-70, 91-93, 96-102, 111-114

System for monitoring usage of software, comprising a software vendor, a tag server and a user device running a supervising program

2. Claims: 19-35, 36-39, 74-90, 108-110, 115-118

Centralizing control at a specific server of software usage at the client

3. Claims: 103-105, 119-131, 134, 40-52, 71-73

Intercepting use of unregistered instances of software

4. Claims: 106, 107, 94, 95

Uniquely identifying instances of software prior distribution

5. Claims: 132-133, 135-137

Limiting use of software on specific hardware devices

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern. Application No

PCT/US 00/11821

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9845768 A	15-10-1998	US 6108420 A AU 6492198 A CN 1255209 T EP 0974084 A	22-08-2000 30-10-1998 31-05-2000 26-01-2000
US 5825883 A	20-10-1998	NONE	
US 5023907 A	11-06-1991	NONE	

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW

(72)発明者 ラビン・マイケル・オー
アメリカ合衆国, マサチューセッツ州
02138, ケンブリッジ, コンコード アベ
ニュー 243

(72)発明者 シャーシャ・デニス・イー
アメリカ合衆国, ニューヨーク州 10012,
ニューヨーク, プリーカー ストリート
100

Fターム(参考) 5B076 FB03 FB05 FB19 FD01
5B085 AE26 BG07